Package 'WaverideR'

April 28, 2025

Type Package

Title Extracting Signals from Wavelet Spectra

Version 0.4.1

Maintainer Michiel Arts <michiel.arts@stratigraphy.eu>

Depends R (>= 3.5.0)

Imports DescTools, Hmisc, Matrix,utils,colorednoise,
foreach,stats,matrixStats,reshape2,truncnorm,grDevices,graphics,parallel,astrochron,RColorBrewer,colorRamps,viridis,ma

Description The continuous wavelet transform enables the observation of transient/non-stationary cyclicity in time-series. The goal of cyclostratigraphic studies is to define frequency/period in the depth/time domain. By conducting the continuous wavelet transform on cyclostratigraphic data series one can observe and extract cyclic signals/signatures from signals. These results can then be visualized and interpreted enabling one to identify/interpret cyclicity in the geological record, which can be used to construct astrochronological age-models and identify and interpret cyclicity in past and present climate systems. The 'WaverideR' R package builds upon existing literature and existing codebase. The list of articles which are relevant can be grouped in four subjects; cyclostratigraphic data analysis,example data sets,the (continuous) wavelet transform and astronomical solutions. References for the cyclostratigraphic data analysis articles are: Stephen Meyers (2019) <doi:10.1016/j.earscirev.2018.11.015>. Mingsong Li, Linda Hinnov, Lee Kump (2019) <doi:10.1016/j.cageo.2019.02.011> Stephen Meyers (2012)<doi:10.1029/2012PA002307> Mingsong Li, Lee R. Kump, Linda A. Hinnov, Michael E. Mann (2018) <doi:10.1016/j.epsl.2018.08.041>. Wouters, S., Crucifix, M., Sinnesael, M., Da Silva, A.C., Zeeden, C., Zivanovic, M., Boulvain, F., Devleeschouwer, X. (2022) <doi:10.1016/j.earscirev.2021.103894>. Wouters, S., Da Silva, A.-C., Boulvain, F., and Devleeschouwer, X. (2021) <doi:10.32614/RJ-2021-039>. Huang, Norden E., Zhaohua Wu, Steven R. Long, Kenneth C. Arnold, Xianyao Chen, and Karin Blank (2009) <doi:10.1142/S1793536909000096>. Cleveland, W. S. (1979)<doi:10.1080/01621459.1979.10481038> Hurvich, C.M., Simonoff, J.S., and Tsai, C.L. (1998) <doi:10.1111/1467-9868.00125>, Golub, G., Heath, M. and Wahba, G. (1979) <doi:10.2307/1268518>. References for the example data articles are: Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu (2018) <doi:10.1016/j.epsl.2018.02.010>. Steinhilber, Friedhelm, Abreu, Jacksiel, Beer, Juerg , Brunner, Irene, Christl, Marcus, Fischer, Hubertus, HeikkilA, U., Kubik, Peter, Mann, Mathias, Mccracken, K. , Miller, Heinrich, Miyahara, Hiroko, Oerter, Hans , Wilhelms, Frank. (2012 <doi:10.1073/pnas.1118965109>. Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert (2013) <doi:10.1016/j.palaeo.2012.11.009>. References for the (continuous) wavelet transform articles are: Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard (1982a) <doi:10.1190/1.1441328>. J. Morlet, G. Arens, E. Fourgeau, D. Giard (1982b) <doi:10.1190/1.1441329>. Torrence, C., and G. P. Compo (1998)<`https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf`>, Gouhier TC, Grinsted A, Simko V (2021) <`https://github.com/tgouhier/biwavelet`>. Angi Roesch and Harald Schmidbauer (2018) <`https://CRAN.R-project.org/package=WaveletComp`>. Russell, Brian, and Jiajun Han (2016)<`https://www.crewes.org/Documents/ResearchReports/2016/CRR201668.pdf`>. Gabor, Dennis (1946) <`http://genesis.eecg.toronto.edu/gabor1946.pdf`>. J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B. (2004) <doi:10.1051/0004-6361:20041335>. Laskar, J., Fienga, A., Gastineau, M., Manche, H. (2011a) <doi:10.1051/0004-6361/201116836>. References for the astronomical solutions articles are: Laskar, J., Gastineau, M., Delisle, J.-B., Farres, A., Fienga, A. (2011b <doi:10.1051/0004-6361/201117504>. J. Laskar (2019) <doi:10.1016/B978-0-12-824360-2.00004-8>. Zeebe, Richard E (2017) <doi:10.3847/1538-3881/aa8cce>. Zeebe, R. E. and Lourens, L. J. (2019) <doi:10.1016/j.epsl.2022.117595>. Richard E. Zeebe Lucas J. Lourens (2022) <doi:10.1126/science.aax0612>.

License GPL (>= 2)

URL `https://github.com/stratigraphy/WaverideR`

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Repository CRAN

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Author Michiel Arts [aut, cre] (<`https://orcid.org/0000-0003-3181-4608`>)

Date/Publication 2025-04-27 23:20:09 UTC

Contents

`add_wavelet` *Add a wavelet plot*

Description

Generates a plot of a wavelet scalogram which can be integrated into a larger composite plot

Usage

```
add_wavelet(
  wavelet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  lower_depth_time = NULL,
  upper_depth_time = NULL,
  n.levels = 100,
  plot.COI = TRUE,
  color_brewer = "grDevices",
  palette_name = "rainbow",
  plot_dir = FALSE,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  plot_horizontal = TRUE,
  period_ticks = 1,
  periodlab = "period (m)",
  main = NULL,
  yaxt = "s",
  xaxt = "s",
  depth_time_lab = "depth (m)"
)
```

Arguments

`wavelet` wavelet object created using the `analyze_wavelet` function.

`lowerPeriod` Lowest period value which will be plotted

`upperPeriod` Highest period value which will be plotted

`lower_depth_time` lowest depth/time value which will be plotted

`upper_depth_time` Highest depth/time value which will be plotted

`n.levels` Number of color levels `Default=100`.

`plot.COI` Option to plot the cone of influence `Default=TRUE`.

`color_brewer` Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "`Default=grDevices`

`palette_name` Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

`plot_dir` The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE `plot_dir=TRUE`

`add_lines` Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values `Default=NULL`

`add_points` Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values `Default=NULL`

`add_abline_h` Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`

`add_abline_v` Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`

`plot_horizontal` plot the wavelet horizontal or vertical eg y axis is depth or y axis power `Default=TRUE`

`period_ticks`	tick mark spacing 1 is all tickmarks and higher value removes tick marks by the fraction of the tick mark spacing value, the opposite is true for value lower than 1 which will add aditional tickmarks
`periodlab`	lable for the the period column
`main`	main title
`yaxt`	turn on of off the yaxis "s" is on "n" is off `Default="s"`
`xaxt`	turn on of off the xaxis "s" is on "n" is off `Default="s"`
`depth_time_lab`	lable for the the depth/time column

Value

returns a plot of a wavelet scalogram

Author(s)

Code based on the "analyze.wavelet" and "wt.image" functions of the 'WaveletComp' R package and the "wt" function of the 'biwavelet' R package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo (1998). The MTM analysis is from the astrochron R package of Meyers et al., (2012)

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#generate a plot for the magnetic susceptibility data set of Pas et al., (2018)

plot.new()
layout.matrix <- matrix(c(rep(0, 2), 1, 0,0,seq(2, 6, by = 1)),
                       nrow = 2,
                      ncol = 5 ,
                       byrow = TRUE)
graphics::layout(mat = layout.matrix,
                heights = c(0.25, 1),
                # Heights of the two rows
```

```
                widths = c(rep(c(1, 2, 4,2,2), 2)))

par(mar = c(0, 0.5, 1, 0.5))


mag_wt <-
 analyze_wavelet(
   data = mag,
   dj = 1 / 100,
  lowerPeriod = 0.1,
   upperPeriod = 254,
   verbose = FALSE,
   omega_nr = 10
 )

 add_wavelet_avg(
 wavelet = mag_wt,
 plot_horizontal = TRUE,
 add_abline_h = NULL,
 add_abline_v = NULL,
 lowerPeriod = 0.15,
 upperPeriod = 80
)

par(mar = c(4, 4, 0, 0.5))


plot(
 x = c(0, 1),
 y = c(max(mag[, 1]), min(mag[, 1])),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1])))
)             # Draw empty plot


polygon(
 x = c(0, 1, 1, 0),
 y = c(max(mag[, 1]), max(mag[, 1]), min(mag[, 1]), min(mag[, 1])),
 col = geo_col("Famennian")
)

text(
 0.5,
 (max(mag[, 1]) - min(mag[, 1])) / 2,
 "Fammenian",
 cex = 1,
 col = "black",
 srt = 90
```

```
)
par(mar = c(4, 0.5, 0, 0.5))


plot(
 mag[, 2],
 mag[, 1],
 type = "l",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1]))),
 yaxs = "i",
 yaxt = "n",
 xlab = "Mag. suc.",
 ylab = ""
)


add_wavelet(
 wavelet = mag_wt,
 lowerPeriod = 0.15,
 upperPeriod = 80,
 lower_depth_time = NULL,
 upper_depth_time = NULL,
 n.levels = 100,
 plot.COI = TRUE,
 color_brewer = "grDevices",
 palette_name = "rainbow",
 plot_dir = FALSE,
 add_lines = NULL,
 add_points = NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 plot_horizontal = TRUE,
 period_ticks = 1,
 periodlab = "period (m)",
 main = NULL,
 yaxt = "n",
 xaxt = "s",
 depth_time_lab = ""
)


lines(log2(mag_track_solution[,2]),mag_track_solution[,1],lwd=4,lty=4)

mag_405 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 wavelet = mag_wt,
 period_up = 1.2,
 period_down = 0.8,
 add_mean = TRUE,
 tracked_cycle_period = 405,
 extract_cycle = 405,
 tune = FALSE,
 plot_residual = FALSE
```

```
)

plot(mag_405[,2],mag_405[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="405-kyr ecc")


mag_110 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 wavelet = mag_wt,
 period_up = 1.25,
 period_down = 0.75,
 add_mean = TRUE,
 tracked_cycle_period = 405,
 extract_cycle = 110,
 tune = FALSE,
 plot_residual = FALSE
)

mag_110_hil <- Hilbert_transform(mag_110,demean=FALSE)

plot(mag_110[,2],mag_110[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="110-kyr ecc")

lines(mag_110_hil[,2],mag_110_hil[,1])
```

---

add_wavelet_avg *Add a plot of a the average spectral power of a continous wavelet transform*

Description

Generates a plot of a the average spectral power of a continous wavelet transform which can be added to a larger composite plot

Usage

```
add_wavelet_avg(
  wavelet = NULL,
  plot_horizontal = TRUE,
  add_abline_h = NULL,
  add_abline_v = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL
)
```

Arguments

- `wavelet` — wavelet object created using the `analyze_wavelet` function.
- `plot_horizontal` — plot the wavelet horizontal or vertical eg y axis is depth or y axis power `Default=TRUE`
- `add_abline_h` — Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`
- `add_abline_v` — Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`
- `lowerPeriod` — Lowest period value which will be plotted
- `upperPeriod` — Highest period value which will be plotted

Value

returns a plot of a the average spectral power of a continuous wavelet transform

Author(s)

Code based on the "analyze.wavelet" and "wt.image" functions of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo (1998). The MTM analysis is from the astrochron R package of Meyers et al., (2012)

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#generate a plot for the magnetic susceptibility data set of Pas et al., (2018)

plot.new()
layout.matrix <- matrix(c(rep(0, 2), 1, 0,0,seq(2, 6, by = 1)),
                        nrow = 2,
                       ncol = 5 ,
                        byrow = TRUE)
graphics::layout(mat = layout.matrix,
```

```
                heights = c(0.25, 1),
                # Heights of the two rows
                widths = c(rep(c(1, 2, 4,2,2), 2)))

par(mar = c(0, 0.5, 1, 0.5))


mag_wt <-
 analyze_wavelet(
   data = mag,
   dj = 1 / 100,
  lowerPeriod = 0.1,
   upperPeriod = 254,
   verbose = FALSE,
   omega_nr = 10
 )

add_wavelet_avg(
 wavelet = mag_wt,
 plot_horizontal = TRUE,
 add_abline_h = NULL,
 add_abline_v = NULL,
 lowerPeriod = 0.15,
 upperPeriod = 80
)


par(mar = c(4, 4, 0, 0.5))


plot(
 x = c(0, 1),
 y = c(max(mag[, 1]), min(mag[, 1])),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1])))
)            # Draw empty plot


polygon(
 x = c(0, 1, 1, 0),
 y = c(max(mag[, 1]), max(mag[, 1]), min(mag[, 1]), min(mag[, 1])),
 col = geo_col("Famennian")
)

text(
 0.5,
 (max(mag[, 1]) - min(mag[, 1])) / 2,
 "Fammenian",
```

```
 cex = 1,
 col = "black",
 srt = 90
)
par(mar = c(4, 0.5, 0, 0.5))


plot(
 mag[, 2],
 mag[, 1],
 type = "l",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1]))),
 yaxs = "i",
 yaxt = "n",
 xlab = "Mag. suc.",
 ylab = ""
)

add_wavelet(
 wavelet = mag_wt,
 lowerPeriod = 0.15,
 upperPeriod = 80,
 lower_depth_time = NULL,
 upper_depth_time = NULL,
 n.levels = 100,
 plot.COI = TRUE,
 color_brewer = "grDevices",
 palette_name = "rainbow",
 plot_dir = FALSE,
 add_lines = NULL,
 add_points = NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 plot_horizontal = TRUE,
 period_ticks = 1,
 periodlab = "period (m)",
 main = NULL,
 yaxt = "n",
 xaxt = "s",
 depth_time_lab = ""
)

lines(log2(mag_track_solution[,2]),mag_track_solution[,1],lwd=4,lty=4)

mag_405 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 wavelet = mag_wt,
 period_up = 1.2,
 period_down = 0.8,
 add_mean = TRUE,
 tracked_cycle_period = 405,
 extract_cycle = 405,
 tune = FALSE,
```

```
 plot_residual = FALSE
)

plot(mag_405[,2],mag_405[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="405-kyr ecc")


mag_110 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 wavelet = mag_wt,
 period_up = 1.25,
 period_down = 0.75,
 add_mean = TRUE,
 tracked_cycle_period = 405,
 extract_cycle = 110,
 tune = FALSE,
 plot_residual = FALSE
)

mag_110_hil <- Hilbert_transform(mag_110,demean=FALSE)

plot(mag_110[,2],mag_110[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="110-kyr ecc")

lines(mag_110_hil[,2],mag_110_hil[,1])
```

`age_model_zeeden` *Age model of Zeeden et al., (2013) for the (154-174m) interval of the IODP 926 grey scale record*

Description

Age model (anchor points) of the IODP 926 grey scale (154-174m) record of Zeeden et al., (2013) Anchored to the eccentricity-tilt-precession model p-0.5t of la 2004.

Details

Column 1: Depth (meters)
Column 2: Age (kyr)

References

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B., 2004, A long term numerical solution for the insolation quantities of the Earth: Astron. Astrophys., Volume 428, 261-285. <doi:10.1051/0004-6361:20041335>

`analyze_wavelet` *Conduct the continuous wavelet transform on a time series/signal*

Description

Compute the continuous wavelet transform (CWT) using a Morlet wavelet

Usage

```
analyze_wavelet(
  data = NULL,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_nr = 8,
  pval = FALSE,
  n_simulations = 10,
  run_multicore = FALSE
)
```

Arguments

`data`	Input data, should be a matrix or data frame in which the first column is depth or time and the second column is proxy record.
`dj`	Spacing between successive scales. The CWT analyses analyses the signal using successive periods which increase by the power of 2 (e.g.2^0=1,2^1=2,2^2=4,2^3=8,2^4=16). To have more resolution in-between these steps the dj parameter exists, the dj parameter specifies how many extra steps/spacing in-between the power of 2 scaled CWT is added. The amount of steps is 1/x with a higher x indicating a smaller spacing. Increasing the increases the computational time of the CWT `Default=1/200`.
`lowerPeriod`	Lowest period to be analyzed `Default=2`. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyze the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.
`upperPeriod`	Upper period to be analyzed `Default=1024`. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyze the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.
`verbose`	Print text `Default=FALSE`.
`omega_nr`	Number of cycles contained within the Morlet wavelet

pval	calculate the P-value `Default=FALSE`. The p-value is based on Monte Carlo modelling runs on surrogate data generated based on autocorrelated noise (red noise) the calculated using a windowed (the window is half the size of the data set) temporal autocorrelation and on shuffling the data set resulting in a random data sets which has similar spectral characteristics to the original data set.The shuffling of the data set creates white noise which ensures that high amplitude high frequency/short period cycles do not result in statistical significant peaks. The part of the data generated using the autocorrelated noise (red noise) based on the windowed (the window is half the size of the data set) temporal autocorrelation represent a spectral signature similar to to that of the original data. The original data might include spectral peaks which are the result of astronomical forcing. The result is that the spectral power profile is biased towards rejecting the 0-hypothesis (e.g. no astronomical forcing). By combining the shuffling of the data set with autocorrelated noise a surrogate data set is created which rejects high amplitude high frequency/short period cycles and a reduced biased towards towards rejecting the 0-hypothesis if the data was solely the result of autocorrelated noise
n_simulations	Number of simulation to be ran to generate the p-value
run_multicore	Run p-value calculation with one core or multiple cores

Value

The output is a list (wavelet object) which contain 20 objects which are the result of the continuous wavelet transform (CWT). Object 1: Wave - Wave values of the wavelet Object 2: Phase - Phase of the wavelet Object 3: Ampl - Amplitude values of the wavelet Object 4: Power - Power values of the wavelet Object 5: dt - Step size Object 6: dj - Scale size Object 7: Power.avg - Average power values Object 8: Period - Period values Object 9: Scale - Scale value Object 10: coi.1 - Cone of influence values 1 Object 11: coi.2 - Cone of influence values 2 Object 12: nc - Number of columns Object 13: nr - Number of rows Object 14: axis.1 - axis values 1 Object 15: axis.2 - axis values 2 Object 16: omega_nr - Number of cycles in the wavelet Object 17: x - x values of the data set Object 18: y - y values of the data set Object 19: average p value of the spectral power Object 20: p value of spectral power

Author(s)

Code based on on the "WaveletComp" function of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo.

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#Example 1. Using the Total Solar Irradiance data set of Steinhilver et al., (2012)
TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6,
   pval=FALSE,
   n_simulations=10,
   run_multicore = FALSE
 )


#Example 2. Using the magnetic susceptibility data set of Pas et al., (2018)
mag_wt <-
analyze_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10,
pval=FALSE,
n_simulations=10,
run_multicore = FALSE
)

#Example 3. Using the greyscale data set of Zeeden et al., (2013)
grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8,
   pval=FALSE,
   n_simulations=10,
   run_multicore = FALSE
 )
```

anchor2time *Convert a proxy record to the time domain using anchor points*

Description

Convert a proxy record to the time domain using anchor points made using the astro_anchor function.

Usage

```
anchor2time(
  anchor_points = NULL,
  data = NULL,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

anchor_points	Anchor points made using the astro_anchor function or a matrix in which the first column is depth and the second column is time.
data	Data set which needs to be converted from the depth to time domain using set anchor points. The data set should consist of a matrix with 2 column the first column should be depth and the second column should be a proxy value.
genplot	If genplot=FALSE then 3 plots stacked on top of each other will be plotted. Plot 1: the original data set Plot 2: the depth time plot Plot 3: the data set in the time domain set to TRUE to allow for anchoring using the GUI
keep_editable	Keep option to add extra features after plotting Default=FALSE

Value

The output is a matrix with 2 columns. The first column is time. The second column sedimentation proxy value.

If genplot=TRUE then 3 plots stacked on top of each other will be plotted. Plot 1: the original data set. Plot 2: the depth time plot. Plot 3: the data set in the time domain.

Examples

```
# Use the age_model_zeeden example anchor points of Zeeden et al., (2013)
#to anchor the grey data set of Zeeden et al., (2013) in the time domain.

grey_time <- anchor2time(anchor_points=age_model_zeeden,
data=grey,
genplot=FALSE,
keep_editable=FALSE)
```

`anchor_points_Bisciaro_al`
XRF records of the Bisciaro Fm

Description

data set consist of the tie points between the Bisciaro_al record of Arts (2014) and the la2011 solution of laskar et al., (20111)

Details

The data set is a matrix with the 4 columns. The first column is the depth/time of the al proxy record tie-points. The second column is the time value of the la2011 astronomical solution tie-points. The third column is the Al value of the a; tie-point. The fourth column is the eccentricity value of the la2011 astronomical solution tie-point.

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

Laskar, J., M. Gastineau, J. B. Delisle, A. Farrés, and A. Fienga (2011b), Strong chaos induced by close encounters with Ceres and Vesta, Astron. Astrophys., 532, L4,<doi:10.1051/0004-6361/201117504>

`anchor_points_grey`
Example anchor points for the grey scale data set of Zeeden et al., (2013)

Description

An example of anchor points generated using `astro_anchor` function
The anchor points were generated for the `grey` grey data set of
Zeeden et al. (2013) and anchored to the `astrosignal_example`
astronomical solution which is a pre-generated ETP (eccentricity-tilt-precession)
solution(p-0.5t based on the la2004 solution) based on Laskar et al., (20004)
astronomical solution.

Details

Column 1: depth proxy record
Column 2: time astronomical solution
Column 3: y-scale value proxy record
Column 4: y-scale value astronomical solution

References

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B., 2004, A long term numerical solution for the insolation quantities of the Earth: Astron. Astrophys., Volume 428, 261-285. <doi:10.1051/0004-6361:20041335>

---

astrosignal_example *An ETP astronomical solution*

Description

The astrosignal_example is a pre-generated ETP
(eccentricity-tilt-precession) (p-0.5t based on the la2004 solution)
the astrosignal_example can be used to anchor the grey
data set to an astronomical solution eg. astrosignal_example
using the astro_anchor function. the data set was generated using the
etp function of the 'astrochron' R package. The pre-generated ETP spans 5000 to 6000kyr.

Details

Column 1: time (kyr)
Column 2: ETP

Author(s)

Generated using the etp function of the astrochron-package.

References

Stephen R. Meyers,Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews,Volume 190,2019,Pages 190-223,ISSN 0012-8252 <doi:10.1016/j.earscirev.2018.11.015>

J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B., 2004, A long term numerical solution for the insolation quantities of the Earth: Astron. Astrophys., Volume 428, 261-285. <doi:10.1051/0004-6361:20041335>

astro_anchor *Anchor proxy record to an astronomical solution*

Description

Anchor the extracted signal to an astronomical solution using a GUI. The astro_anchor function allows one to tie minima or maxima in the proxy record to minima or maxima in an astronomical solution. By tying the proxy record to an astronomical solution one will generate tie-points which can be used to generate a astrochronological age-model As minima or maxima in the proxy record are tied to minima or maxima in an astronomical solution it is important to provide input which has clearly definable minima and maxima. As such input should be of a "sinusoidal" nature otherwise the `extract_astrosolution=TRUE` and/or `extract_proxy_signal=TRUE` options need to be set to TRUE to create sinusoidal signals.

Astronomical solutions option are:

- La2004 Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2004 Obliquity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2004 Precession solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010a Eccentricity solution available via the getLaskarfunction or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010a Obliquity solution downloadable via the http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010a Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010b Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010b Obliquity solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010b Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010c Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010c Obliquity solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010c Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010d Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010d Obliquity solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html

- La2010d Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2011 Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- ZB17a Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17a Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17b Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17b Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17c Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17c Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17d Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17d Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17e Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17e Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17f Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17f Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17h Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17h Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17i Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17i Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17j Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17j Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17k Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html

- ZB17k Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17p Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17p Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB18a Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB18a Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20a Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20a Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20b Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20b Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20c Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20c Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20d Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20d Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- 405kyr eccentricity 405 metronome can be generated using the formula:
  e405=0.027558-0.010739*cos(0.0118+2(pi)*(t/405000)) (laskar et al., 2004 & laskar 2020)
- 173kyr obliquity metronome can be generated using using the formula:
  es3-s6(t) = 0.144*cos(1.961+2(pi)*(t/172800) (laskar et al., 2004 & laskar 2020)
- An etp model using the etp function of the 'astrochron' R package

Usage

```
astro_anchor(
  astro_solution = NULL,
  proxy_signal = NULL,
  proxy_min_or_max = "max",
  clip_astrosolution = FALSE,
  astrosolution_min_or_max = "max",
  clip_high = NULL,
  clip_low = NULL,
  extract_astrosolution = FALSE,
  astro_period_up = 1.2,
```

```
  astro_period_down = 0.8,
  astro_period_cycle = NULL,
  extract_proxy_signal = FALSE,
  proxy_period_up = 1.2,
  proxy_period_down = 0.8,
  proxy_period_cycle = NULL,
  pts = 3,
  verbose = FALSE,
  time_dir = TRUE,
  genplot = FALSE
)
```

Arguments

`astro_solution` Input is an astronomical solution which the proxy record will be anchored to, the input should be a matrix or data frame with the first column being age and the second column should be a insolation/angle/value

`proxy_signal` Input is the proxy data set which will be anchored to an astronomical solution, the input should be a matrix or data frame with the first column being depth/time and the second column should be a proxy value. For the best results either the astronomical components need to be pre-extracted before anchoring. This means that a filtering/cycle extracting need to be applied to the input data or the extract_proxy_signal option needs to be set to TRUE.

`proxy_min_or_max`
Tune proxy maxima or minima to the astronomical solution `Default="max"`.

`clip_astrosolution`
Clip the astronomical solution `Default=FALSE`.

`astrosolution_min_or_max`
Tune to maximum or minimum values of the astronomical solution `Default="max"`

`clip_high` Upper value to clip to.

`clip_low` Lower value to clip to.

`extract_astrosolution`
Extract a certain astronomical cycle/component from a astronomical solution prior to anchoring `Default=FALSE`.

`astro_period_up`
Specifies the upper period of the astronomical cycle which is extracted from an astronomical solution. The astro_period_up is a factor with which the astronomical component is multiplied by. `Default=1.2`

`astro_period_down`
Specified the lower period of the astronomical cycle which is extracted from an astronomical solution. The astro_period_down value is a factor with which the astronomical component is multiplied by. `Default=0.8`

`astro_period_cycle`
Period (in kyr) of the to be extracted astronomical component from the astronomical solution.

`extract_proxy_signal`
Extract a certain astronomical cycle/component from a proxy signal `Default=FALSE`.

`proxy_period_up`
: Specifies the upper period of the astronomical cycle to be extracted from the proxy record. The upper bound value is a factor with which the (proxy_period_cycle) value is multiplied by. `Default=1.2`.

`proxy_period_down`
: Specifies the upper period of the astronomical cycle to be extracted from the proxy record. The lower bound value is a factor with which the astronomical component (proxy_period_cycle) value is multiplied by. `Default=0.8`.

`proxy_period_cycle`
: Period in kyr of the astronomical cycle/component which is to be extracted from the proxy record.

`pts`
: The pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=3`

`verbose`
: print text `Default=FALSE` set verbose to TRUE to allow for anchoring using text feedback commands

`time_dir`
: The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then time_dir should be set to TRUE if time decreases with depth/time values (eg stratigraphic logs where 0m is the bottom of the section) then time_dir should be set to FALSE `time_dir=TRUE`

`genplot`
: Generate plot `Default="FALSE"`

Value

The output is a matrix with the 4 columns. The first column is the depth/time of the proxy tie-point. The second column is the time value of the astronomical solution tie-point. The third column is the proxy value of the proxy tie-point. The fourth column is the proxy/insolation value of the astronomical solution tie-point. If genplot is set to true then at plot of the of the achored points will be plotted

References

J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B., 2004, A long term numerical solution for the insolation quantities of the Earth: Astron. Astrophys., Volume 428, 261-285. <doi:10.1051/0004-6361:20041335>

Laskar, J., Fienga, A., Gastineau, M., Manche, H., 2011a, La2010: A new orbital solution for the long-term motion of the Earth: Astron. Astrophys., Volume 532, A89 <doi:10.1051/0004-6361/201116836>

Laskar, J., Gastineau, M., Delisle, J.-B., Farres, A., Fienga, A.: 2011b, Strong chaos induced by close encounters with Ceres and Vesta, Astron: Astrophys., Volume 532, L4. <doi:10.1051/0004-6361/201117504>

J. Laskar,Chapter 4 - Astrochronology,Editor(s): Felix M. Gradstein, James G. Ogg, Mark D. Schmitz, Gabi M. Ogg,Geologic Time Scale 2020,Elsevier,2020,Pages 139-158,ISBN 9780128243602, <doi:10.1016/B978-0-12-824360-2.00004-8>

Zeebe, R. E. and Lourens, L. J. Geologically constrained astronomical solutions for the Cenozoic era, Earth and Planetary Science Letters, 2022 <doi:10.1016/j.epsl.2022.117595>

Richard E. Zeebe Lucas J. Lourens ,Solar System chaos and the Paleocene–Eocene boundary age constrained by geology and astronomy.Science365,926-929(2019) <doi:10.1126/science.aax0612>

Zeebe, Richard E. "Numerical solutions for the orbital motion of the Solar System over the past 100 Myr: limits and new results." The Astronomical Journal 154, no. 5 (2017): 193. <doi:10.3847/1538-3881/aa8cce>

Stephen R. Meyers,Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews,Volume 190,2019,Pages 190-223,ISSN 0012-8252 <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
# Use the grey_track example tracking points to anchor the grey scale data set
# of Zeeden et al., (2013) to the p-0.5t la2004 solution

grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8
 )
#Use the pre-tracked grey_track curve which traced the precession cycle
grey_track <- completed_series(
 wavelet = grey_wt,
 tracked_curve  = grey_track,
 period_up  = 1.25,
 period_down  = 0.75,
 extrapolate = TRUE,
genplot = FALSE
)
# Extract precession, obliquity and eccentricity to create a synthetic insolation curve

grey_prec <- extract_signal(
tracked_cycle_curve = grey_track[,c(1,2)],
wavelet = grey_wt,
```

```
period_up = 1.2,
period_down = 0.8,
add_mean = FALSE,
tracked_cycle_period = 22,
extract_cycle = 22,
tune = FALSE,
plot_residual = FALSE
)

grey_obl <- extract_signal(
 tracked_cycle_curve = grey_track[,c(1,2)],
 wavelet = grey_wt,
 period_up = 1.2,
 period_down = 0.8,
 add_mean = FALSE,
 tracked_cycle_period = 22,
 extract_cycle = 110,
 tune = FALSE,
 plot_residual = FALSE
)

grey_ecc <- extract_signal(
 tracked_cycle_curve = grey_track[,c(1,2)],
 wavelet = grey_wt,
 period_up = 1.25,
 period_down = 0.75,
 add_mean = FALSE,
 tracked_cycle_period = 22,
 extract_cycle = 40.8,
 tune = FALSE,
 plot_residual = FALSE
)

insolation_extract <- cbind(grey_ecc[,1],grey_prec[,2]+grey_obl[,2]+grey_ecc[,2]+mean(grey[,2]))
insolation_extract <- as.data.frame(insolation_extract)
insolation_extract_mins <- min_detect(insolation_extract,pts=3)

#use the astrosignal_example to tune to which is an \cr
# ETP solution (p-0.5t la2004 solution)
astrosignal_example <- na.omit(astrosignal_example)
astrosignal_example[,2] <- -1*astrosignal_example[,2]
astrosignal <- as.data.frame(astrosignal_example)

#anchor the synthetic insolation curve extracted from the grey scale record to the insolation curve.

anchor_pts <- astro_anchor(
astro_solution = astrosignal,
proxy_signal = insolation_extract,
proxy_min_or_max = "min",
clip_astrosolution = FALSE,
astrosolution_min_or_max = "min",
clip_high = NULL,
clip_low = NULL,
```

```
extract_astrosolution  = FALSE,
astro_period_up  = NULL,
astro_period_down  = NULL,
astro_period_cycle  = NULL,
extract_proxy_signal  = FALSE,
proxy_period_up  = NULL,
proxy_period_down  = NULL,
proxy_period_cycle  = NULL,
pts=3,
verbose=FALSE, #set verbose to TRUE to allow for anchoring using text feedback commands
genplot=FALSE
)
```

Bisciaro_al_wt_track *Period of the short kyr ecc cycle in the Al record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked
in the wavelet scalogram of the aluminium (XRF) record of the Bisciaro Formation
The period was tracked using the track_period_wavelet function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the Aluminium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

Bisciaro_ca_wt_track *Period of the short kyr ecc cycle in the Ca record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked
in the wavelet scalogram of the calcium (XRF) record of the Bisciaro Formation
The period was tracked using the track_period_wavelet function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the calcium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`Bisciaro_Mg_wt_track` *Period of the short kyr ecc cycle in the Mg record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked
in the wavelet scalogram of the magnesium (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the Magnesium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`Bisciaro_Mn_wt_track` *Period of the short kyr ecc cycle in the Mn record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked
in the wavelet scalogram of the manganese (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the manganese (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

Bisciaro_sial_wt_track *Period of the short kyr ecc cycle in the si/Al record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked
in the wavelet scalogram of the silicon/aluminium (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the silicon/aluminium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

Bisciaro_XRF *XRF records of the Bisciaro Fm*

Description

XRF proxy records from the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy)

Details

Column 1: depth proxy record
Column 2-71: XRF proxy records

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

completed_series *Complete the tracking of cycle in a wavelet spectra*

Description

Use the traced series and the existing wavelet spectra to complete the tracking of a cycle of the wavelet spectra. The selected points using the `track_period_wavelet` function form a incomplete line unless every point is tracked. However clicking every individual point along a wavelet ridge is time intensive and error prone. To avoid errors and save time the `completed_series` function can be used to complete the tracing of a cycle in a wavelet spectra.The `completed_series` function interpolates the data points selected using the `track_period_wavelet`. A a search a algorithm then looks up and replaces the interpolated curve values with the values of the nearest spectral peak in the wavelet spectra.

Usage

```
completed_series(
  wavelet = NULL,
  tracked_curve = NULL,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

`wavelet`	Wavelet object created using the `analyze_wavelet` function.
`tracked_curve`	Traced period result from the `track_period_wavelet` function.
`period_up`	The period_up parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_up factor is the upper boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks `Default=1.2`,
`period_down`	The period_down parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_down factor is the lower boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks `Default=0.8`,

extrapolate	Extrapolate the completed curve when through parts where no spectral peaks could be traced Default=TRUE.
genplot	Generate a plot Default=TRUE. The red curve is the completed curve, the black curve is the original curve.
keep_editable	Keep option to add extra features after plotting Default=FALSE

Value

Returns a matrix with 2 columns The first column is the depth axis The second column is the completed tracking of the period a cycle of the wavelet spectra

Examples

```
#Use the grey_track example points to complete the tracking of the
# precession cycle in the wavelet spectra of the grey scale data set
# of Zeeden et al., (2013).



grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8
 )

#The ~22kyr precession cycle is between 0.25 and 1m The grey_track data
#set is a pre-loaded uncompleted tracking of the precession cycle

#grey_track <- track_period_wavelet(
#astro_cycle = 22,
#wavelet = NULL,
#n.levels = 100,
#periodlab = "Period (meters)",
#x_lab = "depth (meters)"
#)



grey_track <- completed_series(
 wavelet = grey_wt,
 tracked_curve  = grey_track,
 period_up  = 1.25,
 period_down  = 0.75,
 extrapolate = TRUE,
genplot = FALSE,
keep_editable=FALSE
)
```

curve2sedrate *Convert a tracked tracked to a sedimentation rate curve*

Description

Converts the period of a tracked cycle to a sedimentation rate curve by assigning a duration (in kyr) to the period of a tracked cycle

Usage

```
curve2sedrate(tracked_cycle_curve = NULL, tracked_cycle_period = NULL)
```

Arguments

`tracked_cycle_curve`
: A tracked cycle which is the result of using the `track_period_wavelet` function
Any input (matrix or data frame) in which the first column is depth in meters and the second column is period in meters

`tracked_cycle_period`
: Period of the tracked cycle (in kyr).

Value

The output is a matrix with 2 columns The first column is depth The second column sedimentation rate in cm/kyr

Examples

```
#Conversion of the period (in meters) of a 405 kyr eccentricity cycle tracked
#in a wavelet spectra by assigning a duration of  405 kyr to the tracked cycle.
# the example uses the magnetic susceptibility data set of Pas et al., (2018)
# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)")
```

```
#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)

#convert period in meters to sedrate in cm/kyr
mag_track_sedrate <- curve2sedrate(tracked_cycle_curve=mag_track_complete,
tracked_cycle_period=405)
```

---

curve2time *Convert the tracked curve to a depth time space*

Description

Converts the tracked curve to a depth time space.

Usage

```
curve2time(
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

`tracked_cycle_curve`
: Curve of the cycle tracked using the `track_period_wavelet` function
Any input (matrix or data frame) in which the first column is depth in meters and the second column is period in meters can be used.

`tracked_cycle_period`
: Period of the tracked curve in kyr.

`genplot`
: Generates a plot with depth vs time `Default=FALSE`.

`keep_editable`
: Keep option to add extra features after plotting `Default=FALSE`

Value

The output is a matrix with 2 columns. The first column is depth. The second column sedimentation rate in cm/kyr. If genplot=TRUE then a depth vs time plot will be plotted.

Author(s)

Based on the sedrate2time function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#Convert a tracked curve to a depth time space. The examples uses the
#magnetic susceptibility data set of Pas et al., (2018).

#'# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set mag_track_solution is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)

#convert period in meters to sedrate depth vs time
mag_track_time<- curve2time(tracked_cycle_curve=mag_track_complete,
```

```
tracked_cycle_period=405,
genplot=FALSE,
keep_editable=FALSE)
```

---

curve2time_unc — *Convert the re-tracked curve results to a depth time space with uncertainty*

Description

Converts the re-tracked curve results from `retrack_wt_MC` function to a depth time space while also taking into account the uncertainty of the tracked astronomical cycle

Usage

```
curve2time_unc(
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  tracked_cycle_period_unc = NULL,
  tracked_cycle_period_unc_dist = "n",
  n_simulations = NULL,
  output = 1
)
```

Arguments

`tracked_cycle_curve`
: Curve of the cycle tracked using the `retrack_wt_MC` function
Any input (matrix or data frame) with 3 columns in which column 1 is the x-axis, column 2 is the mean tracked frequency (in cycles/metres) column 3 1 standard deviation

`tracked_cycle_period`
: Period of the tracked curve in kyr.

`tracked_cycle_period_unc`
: uncertainty in the period of the tracked cycle

`tracked_cycle_period_unc_dist`
: distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution `Default="n"`

`n_simulations`
: number of time series to be modeled

`output`
: If output = 1 a matrix with the predicted ages given the input for each run is given. If output = 2 a matrix with 6 columns is generated, the first column is depth/height, the other columns are the quantile (0.025,0.373,0.5,0.6827,0.975) age values of the runs if output = 3 a matrix with 4 columns is generated with the first column being depth/height, column 2 is the mean tracked duration (in kyrs) column 3 is mean duration + 1 standard deviation and column 4 is mean duration - 1 standard deviation

Value

If output = 1 a matrix with the predicted ages given the input for each run is given If output = 2 a matrix with 6 columns is generated, the first column is depth/height, the other columns are the quantile (0.02275, 0.373, 0.5, 0.6827, 0.97725) age values of the runs if output = 3 a matrix with 4 columns is generated with the first column being depth/height, column 2 is the mean tracked duration (in kyrs) column 3 is mean duration + 1 standard deviation and column 4 is mean duration - 1 standard deviation

Author(s)

Based on the sedrate2time function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
# Re-track the 110kyr eccentricity cycle in the wavelet scalogram
# from the XRF record of the Bisciaro data set of Arts (2014) and then
# add generate and age model including uncertainty

Bisciaro_al <- Bisciaro_XRF[, c(1, 61)]
Bisciaro_al <- astrochron::sortNave(Bisciaro_al,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- astrochron::linterp(Bisciaro_al, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- Bisciaro_al[Bisciaro_al[, 1] > 2, ]

Bisciaro_al_wt <-
 analyze_wavelet(
   data = Bisciaro_al,
   dj = 1 /200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_al_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_al_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_al_wt_track <- completed_series(
#   wavelet = Bisciaro_al_wt,
#   tracked_curve = Bisciaro_al_wt_track,
#   period_up = 1.2,
```

```
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_al_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_al_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_ca <- Bisciaro_XRF[, c(1, 55)]
Bisciaro_ca <- astrochron::sortNave(Bisciaro_ca,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- astrochron::linterp(Bisciaro_ca, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- Bisciaro_ca[Bisciaro_ca[, 1] > 2, ]

Bisciaro_ca_wt <-
 analyze_wavelet(
   data = Bisciaro_ca,
   dj = 1 /200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_ca_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_ca_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_ca_wt_track <- completed_series(
#   wavelet = Bisciaro_ca_wt,
#   tracked_curve = Bisciaro_ca_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_ca_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_ca_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
```

```
#     keep_editable = FALSE)

Bisciaro_sial <- Bisciaro_XRF[,c(1,64)]
Bisciaro_sial <- astrochron::sortNave(Bisciaro_sial,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- astrochron::linterp(Bisciaro_sial, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- Bisciaro_sial[Bisciaro_sial[, 1] > 2, ]

Bisciaro_sial_wt <-
 analyze_wavelet(
   data = Bisciaro_sial,
   dj = 1 /200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_sial_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_sial_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_sial_wt_track <- completed_series(
#   wavelet = Bisciaro_sial_wt,
#   tracked_curve = Bisciaro_sial_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_sial_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_sial_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )


Bisciaro_Mn <- Bisciaro_XRF[,c(1,46)]
Bisciaro_Mn <- astrochron::sortNave(Bisciaro_Mn,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- astrochron::linterp(Bisciaro_Mn, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- Bisciaro_Mn[Bisciaro_Mn[, 1] > 2, ]

Bisciaro_Mn_wt <-
 analyze_wavelet(
```

```
    data = Bisciaro_Mn,
    dj = 1 /200 ,
    lowerPeriod = 0.01,
    upperPeriod = 50,
    verbose = FALSE,
    omega_nr = 8
  )

# Bisciaro_Mn_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mn_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mn_wt_track <- completed_series(
#   wavelet = Bisciaro_Mn_wt,
#   tracked_curve = Bisciaro_Mn_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
# Bisciaro_Mn_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mn_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_Mg <- Bisciaro_XRF[,c(1,71)]
Bisciaro_Mg <- astrochron::sortNave(Bisciaro_Mg,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- astrochron::linterp(Bisciaro_Mg, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- Bisciaro_Mg[Bisciaro_Mg[, 1] > 2, ]

Bisciaro_Mg_wt <-
 analyze_wavelet(
    data = Bisciaro_Mg,
    dj = 1 /200 ,
    lowerPeriod = 0.01,
    upperPeriod = 50,
    verbose = FALSE,
    omega_nr = 8
  )

# Bisciaro_Mg_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
```

```
#     wavelet = Bisciaro_Mg_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mg_wt_track <- completed_series(
#   wavelet = Bisciaro_Mg_wt,
#   tracked_curve = Bisciaro_Mg_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_Mg_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mg_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE)




wt_list_bisc <- list(Bisciaro_al_wt,
               Bisciaro_ca_wt,
               Bisciaro_sial_wt,
               Bisciaro_Mn_wt,
               Bisciaro_Mg_wt)

#Instead of tracking, the tracked solution data sets Bisciaro_al_wt_track,
#Bisciaro_ca_wt_track, Bisciaro_sial_wt_track, Bisciaro_Mn_wt_track,
# Bisciaro_Mn_wt_track and Bisciaro_Mg_wt_track are used

data_track_bisc <- cbind(Bisciaro_al_wt_track[,2],
                     Bisciaro_ca_wt_track[,2],
                     Bisciaro_sial_wt_track[,2],
                     Bisciaro_Mn_wt_track[,2],
                     Bisciaro_Mg_wt_track[,2])

x_axis_bisc <- Bisciaro_al_wt_track[,1]


bisc_retrack <- retrack_wt_MC(wt_list = wt_list_bisc,
             data_track = data_track_bisc,
             x_axis = x_axis_bisc,
             nr_simulations = 20,
             seed_nr = 1337,
             verbose = FALSE,
             genplot = FALSE,
```

```
                keep_editable = FALSE,
                create_GIF = FALSE,
                plot_GIF = FALSE,
                width_plt =  600,
                height_plt = 450,
               period_up  =  1.5,
                period_down = 0.5,
                plot.COI = TRUE,
                n.levels = 100,
                palette_name = "rainbow",
                color_brewer = "grDevices",
                periodlab = "Period (metres)",
                x_lab = "depth (metres)",
                add_avg = FALSE,
                time_dir = TRUE,
                file_name = NULL,
                run_multicore = FALSE,
                output = 5,
                n_imgs = 50,
                plot_horizontal = TRUE,
                empty_folder = FALSE)

bisc_retrack_age_incl_unc <- curve2time_unc(tracked_cycle_curve = bisc_retrack,
tracked_cycle_period = 110,
tracked_cycle_period_unc = ((135-110)+(110-95))/2,
tracked_cycle_period_unc_dist = "n",
n_simulations = 20,
output = 1)
```

---

curve2time_unc_anchor *Convert the re-tracked curve results to a depth time space with uncertainty*

Description

Converts the re-tracked curve results from retrack_wt_MC function to a depth time space using an anchor date while also taking into account the uncertainty of the tracked astronomical cycle

Usage

```
curve2time_unc_anchor(
  age_constraint = NULL,
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  tracked_cycle_period_unc = NULL,
  tracked_cycle_period_unc_dist = "n",
```

```
  n_simulations = 20,
  gap_constraints = NULL,
  proxy_data = NULL,
  cycles_check = NULL,
  uncer_cycles_check = NULL,
  max_runs = 1000,
  run_multicore = FALSE,
  verbose = FALSE,
  genplot = FALSE,
  keep_nr = 2,
  keep_all_time_curves = FALSE,
  dj = 1/200,
  lowerPeriod = 1,
  upperPeriod = 4600,
  omega_nr = 6,
  seed_nr = 1337,
  dir = TRUE
)
```

Arguments

`age_constraint` age constrains for the modelling run Input should be a data frame with 7 columns, the first columns are the ID names the second column are the ages (usually in kyr) the third column is the uncertainty (usually in kyr) given as the fourth column is the distribution which is either "n" for a normal distribution or "u" for a uniform distribution. The fifth column is the location in the depth domain of the age constraint. the sixth column is the location/thickness uncertainty of the age_constraint in the depth domain. The seventh column is the uncertainty distribution of the age_constrain in the depth domain

`tracked_cycle_curve`
Curve of the cycle tracked using the `retrack_wt_MC` function
Any input (matrix or data frame) with 3 columns in which column 1 is the x-axis, column 2 is the mean tracked frequency (in cycles/metres) column 3 1 standard deviation

`tracked_cycle_period`
Period of the tracked curve in kyr.

`tracked_cycle_period_unc`
uncertainty in the period of the tracked cycle

`tracked_cycle_period_unc_dist`
distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution `Default="n"`

`n_simulations` number of time series to be modeled `Default=20`

`gap_constraints`
gap parameters for the modelling run input should be a data frame with

`proxy_data` proxy data to be tune and check preservation of astronomical cycles

`cycles_check` astronomical cycles which are checked for their presence after tuning

uncer_cycles_check
: uncertainty of astronomical cycles to be check for after tunning

max_runs
: maximum runs before one of the age constraints is dropped `Default=1000`

run_multicore
: Run function using multiple cores `Default="FALSE"`

verbose
: Print text `Default=FALSE`.

genplot
: generate plot code`Default=FALSE`

keep_nr
: minimal number of age constraints to be kept `Default=2`

keep_all_time_curves
: weather to keep all the generated age curves including the ones rejected from the modelling run `Default=FALSE`

dj
: Spacing between successive scales. The CWT analyses analyses the signal using successive periods which increase by the power of 2 (e.g.2^0=1,2^1=2,2^2=4,2^3=8,2^4=16). To have more resolution in-between these steps the dj parameter exists, the dj parameter specifies how many extra steps/spacing in-between the power of 2 scaled CWT is added. The amount of steps is 1/x with a higher x indicating a smaller spacing. Increasing the increases the computational time of the CWT `Default=1/200`.

lowerPeriod
: Lowest period to be analyzed `Default=2`. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyze the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.

upperPeriod
: Upper period to be analyzed `Default=1024`. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyze the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.

omega_nr
: Number of cycles contained within the Morlet wavelet

seed_nr
: The seed number of the Monte-Carlo simulations. `Default=1337`

dir
: time direction of tuning e.g. does time increase or decrease with depth

Value

The output is a list of 3 or 4 elements if keep_all_time_curves is set to TRUE then the list consist of the x-axis, all the fitted curves in a matrix format, the astrochronologically fitted age of the anchor, all the generated depth time curves if keep_all_time_curves is set to TRUE then the list consists of the x-axis, all the fitted curves in a matrix format and the astrochronologically fitted age of the anchor If `genplot=TRUE` then 3 plots stacked on top of each other will be plotted. Plot 1: the original data set. Plot 2: the depth time plot. Plot 3: the data set in the time domain. #'

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
## Not run:


Bisciaro_al <- Bisciaro_XRF[, c(1, 61)]
Bisciaro_al <-
 astrochron::sortNave(Bisciaro_al, verbose = FALSE, genplot = FALSE)
Bisciaro_al <-
 astrochron::linterp(Bisciaro_al,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
Bisciaro_al <- Bisciaro_al[Bisciaro_al[, 1] > 2, ]

Bisciaro_al_wt <-
 analyze_wavelet(
   data = Bisciaro_al,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )
# Bisciaro_al_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_al_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_al_wt_track <- completed_series(
#   wavelet = Bisciaro_al_wt,
#   tracked_curve = Bisciaro_al_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_al_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_al_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )
```

```
Bisciaro_ca <- Bisciaro_XRF[, c(1, 55)]
Bisciaro_ca <-
 astrochron::sortNave(Bisciaro_ca, verbose = FALSE, genplot = FALSE)
Bisciaro_ca <-
 astrochron::linterp(Bisciaro_ca,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
Bisciaro_ca <- Bisciaro_ca[Bisciaro_ca[, 1] > 2, ]

Bisciaro_ca_wt <-
 analyze_wavelet(
   data = Bisciaro_ca,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )


#
# Bisciaro_ca_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_ca_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_ca_wt_track <- completed_series(
#   wavelet = Bisciaro_ca_wt,
#   tracked_curve = Bisciaro_ca_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_ca_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_ca_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )


Bisciaro_sial <- Bisciaro_XRF[, c(1, 64)]
Bisciaro_sial <-
 astrochron::sortNave(Bisciaro_sial, verbose = FALSE, genplot = FALSE)
```

```
Bisciaro_sial <-
 astrochron::linterp(Bisciaro_sial,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
Bisciaro_sial <- Bisciaro_sial[Bisciaro_sial[, 1] > 2, ]

Bisciaro_sial_wt <-
 analyze_wavelet(
   data = Bisciaro_sial,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

#Bisciaro_sial_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_sial_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_sial_wt_track <- completed_series(
#   wavelet = Bisciaro_sial_wt,
#   tracked_curve = Bisciaro_sial_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_sial_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_sial_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )


Bisciaro_Mn <- Bisciaro_XRF[, c(1, 46)]
Bisciaro_Mn <-
 astrochron::sortNave(Bisciaro_Mn, verbose = FALSE, genplot = FALSE)
Bisciaro_Mn <-
 astrochron::linterp(Bisciaro_Mn,
                     dt = 0.01,
                     verbose = FALSE,
```

```
                    genplot = FALSE)
Bisciaro_Mn <- Bisciaro_Mn[Bisciaro_Mn[, 1] > 2, ]

Bisciaro_Mn_wt <-
 analyze_wavelet(
   data = Bisciaro_Mn,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mn_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mn_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mn_wt_track <- completed_series(
#   wavelet = Bisciaro_Mn_wt,
#   tracked_curve = Bisciaro_Mn_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
# Bisciaro_Mn_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mn_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_Mg <- Bisciaro_XRF[, c(1, 71)]
Bisciaro_Mg <-
 astrochron::sortNave(Bisciaro_Mg, verbose = FALSE, genplot = FALSE)
Bisciaro_Mg <-
 astrochron::linterp(Bisciaro_Mg,
                    dt = 0.01,
                    verbose = FALSE,
                    genplot = FALSE)
Bisciaro_Mg <- Bisciaro_Mg[Bisciaro_Mg[, 1] > 2, ]

Bisciaro_Mg_wt <-
 analyze_wavelet(
   data = Bisciaro_Mg,
```

```
    dj = 1 / 200 ,
    lowerPeriod = 0.01,
    upperPeriod = 50,
    verbose = FALSE,
    omega_nr = 8
  )

# Bisciaro_Mg_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mg_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mg_wt_track <- completed_series(
#   wavelet = Bisciaro_Mg_wt,
#   tracked_curve = Bisciaro_Mg_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_Mg_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mg_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )




wt_list_bisc <- list(Bisciaro_al_wt,
                    Bisciaro_ca_wt,
                    Bisciaro_sial_wt,
                    Bisciaro_Mn_wt,
                    Bisciaro_Mg_wt)


data_track_bisc <- cbind(
 Bisciaro_al_wt_track[, 2],
 Bisciaro_ca_wt_track[, 2],
Bisciaro_sial_wt_track[, 2],
 Bisciaro_Mn_wt_track[, 2],
 Bisciaro_Mg_wt_track[, 2]
)
```

```
x_axis_bisc <- Bisciaro_al_wt_track[, 1]

bisc_retrack <- retrack_wt_MC(
wt_list = wt_list_bisc,
 data_track = data_track_bisc,
 x_axis = x_axis_bisc,
 nr_simulations = 500,
 seed_nr = 1337,
 verbose = TRUE,
 genplot = FALSE,
 keep_editable = FALSE,
 create_GIF = FALSE,
 plot_GIF = FALSE,
 width_plt =  600,
 height_plt = 450,
 period_up  =  1.5,
 period_down = 0.5,
 plot.COI = TRUE,
 n.levels = 100,
 palette_name = "rainbow",
 color_brewer = "grDevices",
periodlab = "Period (metres)",
x_lab = "depth (metres)",
add_avg = FALSE,
time_dir = TRUE,
file_name = "TEST",
run_multicore = TRUE,
output = 5,
 n_imgs = 50,
 plot_horizontal = TRUE,
 empty_folder = FALSE
)

proxy_list_bisc <- list(Bisciaro_al,
                    Bisciaro_ca,
                    Bisciaro_sial,
                    Bisciaro_Mn,
                    Bisciaro_Mg)




id <- c("CCT18_322", "CCT18_315", "CCT18_311")
ages <- c(20158, 20575, 20857)
ageSds <- c(28, 40, 34)
ages_unc_dist <- c("n", "n", "n")
position <- c(13.3, 7.25, 3.2)
anchor_thick <- c(0.2, 0.1, 0.1)
anchor_thick_unc_dist <- c("u", "u", "u")

ash_Bisc <-
 as.data.frame(
   cbind(
```

```
      id,
      ages,
      ageSds,
      ages_unc_dist,
      position,
      anchor_thick,
      anchor_thick_unc_dist
    )
 )

gap_dur = c(10, 20)
gap_unc = c(3, 10)
gap_depth = c(2.5, 9)
gap_unc_dist = c("n", "n")


gap_constraints_Bisc <-
 as.data.frame(cbind(gap_dur, gap_unc, gap_depth, gap_unc_dist))

cycles_checks <- c(110,40,22)
uncer_cycles_checks <- c(20,5,7)

curve2time_unc_anchor_res <-
 curve2time_unc_anchor(
 age_constraint =  ash_Bisc,
  tracked_cycle_curve =  bisc_retrack,
  tracked_cycle_period = 110,
   tracked_cycle_period_unc = ((135 - 110) + (110 - 95)) / 2,
  tracked_cycle_period_unc_dist = "n",
   n_simulations = 20,
   gap_constraints = gap_constraints_Bisc,
   proxy_data = proxy_list_bisc,
   cycles_check = NULL,
   uncer_cycles_check = NULL,
   cycles_check = cycles_checks,
   uncer_cycles_check = uncer_cycles_checks,
  max_runs = 1000,
   run_multicore = FALSE,
   verbose = FALSE,
   genplot = FALSE,
   keep_nr = 2,
   keep_all_time_curves = FALSE,
   dj = 1/200,
   lowerPeriod =1,
   upperPeriod =2500,
   omega_nr = 6,
   seed_nr=1337,
   dir=TRUE
 )


## End(Not run)
```

curve2tune — *Convert data from the depth to the time domain*

Description

Converts a data set from the depth to the time domain using a tracked curve/cycle to depth domain an assigning a duration (in kyr) set tracked curve/cycle.

Usage

```
curve2tune(
  data = NULL,
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

- `data` — Data set (matrix with 2 columns 1st column depth 2nd column proxy value) which was used as input for the `analyze_wavelet` function.
  That result was then used to tracked a cycle using the `track_period_wavelet` function
- `tracked_cycle_curve` — Tracking result of a cycle tracked using the `track_period_wavelet` function Any input (matrix or data frame) in which the first column is depth in meters and the second column is period in meters can be used.
- `tracked_cycle_period` — Period of the tracked curve (in kyr).
- `genplot` — If `genplot=TRUE` 3 plots stacked on top of each other will be plotted. Plot 1: the original data set. Plot 2: the depth time plot. Plot 3: the data set in the time domain.
- `keep_editable` — Keep option to add extra features after plotting `Default=FALSE`

Value

The output is a matrix with 2 columns. The first column is time. The second column sedimentation proxy value.

If `genplot=TRUE` then 3 plots stacked on top of each other will be plotted. Plot 1: the original data set. Plot 2: the depth time plot. Plot 3: the data set in the time domain.

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#The example uses the magnetic susceptibility data set of Pas et al., (2018).
# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                  wavelet=mag_wt,
#                                  n.levels = 100,
#                                  periodlab = "Period (meters)",
#                                  x_lab = "depth (meters)")

#Instead of tracking, the tracked solution data set mag_track_solution is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)

mag_track_time<- curve2tune(data=mag,
                           tracked_cycle_curve=mag_track_complete,
                           tracked_cycle_period=405,
                           genplot = FALSE,
                           keep_editable=FALSE)
```

`delpts_tracked_period_wt`

Remove tracking points which were tracked in a wavelet spectra

---

Description

Interactively select points for deletion With the track_period_wavelet function it is possible to track points in a wavelet spectra, however errors can be made and as such it is possible to delete these points with the delpts_tracked_period_wt function. This function allows one to select points for deletion. #'

Usage

```
delpts_tracked_period_wt(
  tracking_pts = NULL,
  wavelet = NULL,
  n.levels = 100,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  palette_name = "rainbow",
  color_brewer = "grDevices"
)
```

Arguments

tracking_pts	Points tracked using the track_period_wavelet function.
wavelet	Wavelet object created using the analyze_wavelet function.
n.levels	Number of color levels Default=100.
periodlab	label for the y-axis Default="Period (metres)".
x_lab	label for the x-axis Default="depth (metres)".
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices

Value

The results of the deletion of the tracking points is a matrix with 3 columns. The first column is depth/time The second column is the period of the tracked cycle The third column is the sedimentation rate based on the duration (in time) of the tracked cycle

Examples

```
#Track the 405kyr eccentricity cycle in the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)"
#                                    palette_name ="rainbow",
#                                    color_brewer ="grDevices)

#load the mag_track_solution data set to get an example data set from which
#data points can be deleted


mag_track_corr <- delpts_tracked_period_wt(tracking_pts = mag_track_solution,
                                    wavelet = mag_wt,
                                    n.levels = 100,
                                    periodlab = "Period (metres)",
                                    x_lab = "depth (metres)",
                                   palette_name ="rainbow",
                                   color_brewer ="grDevices")
```

---

depth_rank_example *An example depth rank series*

Description

The depth_rank_example example data set is a depth rank series which can be used as input for the lithlog_disc function which creates a discritzed record which can then be used as input in the analyze_wavelet function

Details

Column 1: depth (meters)
Column 2: depth rank

`dur_gaps` *calculate the duration of stratigraphic gaps using astronomical cycles*

Description

calculate the duration of stratigraphic gaps using the duration of stable astronomical cycles

Usage

```
dur_gaps(
  proxies = NULL,
  retracked_period_1 = NULL,
  retracked_period_2 = NULL,
  min_max = NULL,
  n_simulations = 10,
  tracked_cycle_period = NULL,
  tracked_cycle_period_unc = NULL,
  tracked_cycle_period_unc_dist = "u",
  pts = 5,
  dj = 1/200,
  lowerPeriod = 1,
  upperPeriod = 3200,
  period_max = NULL,
  period_min = NULL,
  missing_cycle_dur = NULL,
  n_cycles_missing = 1,
  missing_cycle_unc = NULL,
  missing_cycle_unc_dist = "u",
  seed_nr = 1337,
  run_multicore = FALSE
)
```

Arguments

`proxies` list of proxies which were used to create a astrochronological age model and which are used to calculate the duration of the gap

`retracked_period_1` A matrix of 3 columns in which the first column is depth/height.The second column is the period of the tracked cycle. The thirds column is uncertainty given as 1 standard deviation for the period of the tracked cycle. The gap to be modeled should be located in between retracked_period_1 and retracked_period_2

`retracked_period_2`
: A matrix of 3 columns in which the first column is depth/height.The second column is the period of the tracked cycle. The thirds column is uncertainty given as 1 standard deviation for the period of the tracked cycle. The gap to be modeled should be located in between retracked_period_1 and retracked_period_2

`min_max`
: list of "min" or "max" indicating whether time should be estimated between minima or maxima for each proxy

`n_simulations`
: number of gap duration to calculate

`tracked_cycle_period`
: period in time of the tracked cycle

`tracked_cycle_period_unc`
: uncertainty in the period of the tracked cycle

`tracked_cycle_period_unc_dist`
: distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution `Default="u"`

`pts`
: the pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=5#`'

`dj`
: Spacing between successive scales. The CWT analyses analyses the signal using successive periods which increase by the power of 2 (e.g.2^0=1,2^1=2,2^2=4,2^3=8,2^4=16). To have more resolution in-between these steps the dj parameter exists, the dj parameter specifies how many extra steps/spacing in-between the power of 2 scaled CWT is added. The amount of steps is 1/x with a higher x indicating a smaller spacing. Increasing the increases the computational time of the CWT `Default=1/200`.

`lowerPeriod`
: Lowest period to be analyzed `Default=2`. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyze the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.

`upperPeriod`
: Upper period to be analyzed `Default=1024`. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyze the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.

`period_max`
: Maximum period (upper boundary) to be used to extract a cycle.

`period_min`
: Minimum period (lower boundary) to be used to extract a cycle.

`missing_cycle_dur`
: duration of the missing cycles

`n_cycles_missing`
: number of missing cycles `Default=1`

`missing_cycle_unc`
: duration uncertainty of the missing cycle

missing_cycle_unc_dist
: distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution `Default="u"`

seed_nr
: The seed number of the Monte-Carlo simulations. `Default=1337`

run_multicore
: Run function using multiple cores `Default="FALSE"`

Value

a vector with all the calculated gap durations

`dynamic_extraction` *Extract a signal in between tracked boundaries in a wavelet scalogram*

Description

Interactively select points in a wavelet scalogram to trace the upper and lower period of an cycle. The `dynamic_extraction` function plots a wavelet scalogram in which points peaks can selected allowing one to track the lower and upper period of a cycle. First track the upper or lower period of the to be extracted cycle and then track the other boundary. Tracking points can be selected in the Interactive interface and will be shown as white dots connected by a black line. When one wants to deselect a point the white dots can be re-clicked/re-selected and will turn red which indicates that the previously selected point is deselected. Deselecting points can be quite tricky. After tracking the lower and upper boundaries of the cycle the `dynamic_extraction` function will extract the signal in between the boundaries. the output can then used as input for the `minimal_tuning` function to create an age model.

Usage

```
dynamic_extraction(
  wavelet = NULL,
  n.levels = 100,
  add_peaks = FALSE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_horizontal = TRUE,
  smooth = FALSE,
  add_mean = TRUE
)
```

Arguments

wavelet
: Wavelet object created using the `analyze_wavelet` function.

n.levels
: Number of color levels `Default=100`.

add_peaks
: Setting which indicates whether spectral peaks should be added to the tracking plot `Default=FALSE`.

periodlab — label for the y-axis Default="Period (metres)".

x_lab — label for the x-axis Default="depth (metres)".

palette_name — Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

color_brewer — Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices

plot_horizontal — plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE

smooth — smooth the tracked period using the "loess_auto" function

add_mean — add the mean to the extracted signal

Value

Results of the tracking of a cycle in the wavelet spectra is a matrix with 3 columns. The first column is depth/time The second column is the extracted tracked cycle The third column is upper tracked period The fourth column is lower tracked period

Author(s)

The function is based/inspired on the traceFreq function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
## Not run:
#Track the 405kyr upper and lower periods of the eccentricity cycle in the
#magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

mag_wt <- analyze_wavelet(
```

```
 data = mag,
 dj = 1 / 100,
 lowerPeriod = 0.1,
 upperPeriod = 254,
 verbose = FALSE,
 omega_nr = 10
)

mag_ext <- dynamic_extraction(
 wavelet = mag_wt,
 n.levels = 100,
 add_peaks = FALSE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)",
 palette_name = "rainbow",
 color_brewer = "grDevices",
 plot_horizontal = TRUE,
 smooth = TRUE,
 add_mean = TRUE
)

## End(Not run)
```

---

extract_amplitude *Extract amplitude from a signal*

Description

Extracts the amplitude from a signal using the continuous wavelet transform using a Morlet wavelet. The extraction of the amplitude is useful for cyclostratigraphic studies because the amplitude of an astronomical cycle is modulated by higher order astronomical cycles.

Usage

```
extract_amplitude(
  signal = NULL,
  pts = 3,
  genplot = FALSE,
  remean = TRUE,
  ver_results = FALSE,
  keep_editable = FALSE
)
```

Arguments

signal Input signal from which the amplitude is extracted any signal in which the first column is depth/time and the second column is the proxy record from which the amplitude is extracted

pts	The pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=3`
genplot	If set to TRUE a plot with extracted amplitude will be displayed `Default=FALSE`.
remean	Prior to analysis the mean is subtracted from the data set to re-mean set `Default=TRUE`.
ver_results	To verify the amplitude extraction is representative of the amplitude extracted using the `extract_amplitude` function the results can be compared to the amplitude extracted using the `Hilbert_transform` if the mean difference is more then 5 whether the input contains a reliable enough signal with high a enough amplitude modulation to actually extract an amplitude from. `Default=FALSE`.
keep_editable	Keep option to add extra features after plotting `Default=FALSE`

Value

Returns a matrix with 2 columns. The first column is depth/time. The second column is the extracted amplitude

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo. The assignment of the standard deviation of the uncertainty of the wavelet is based on the work of Gabor (1946) and Russell et al., (2016)

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#Extract amplitude of the 405 kyr eccentricity cycle from the the magnetic
# susceptibility data set of De pas et al., (2018)
```

```
#Perform the CWT on the magnetic susceptibility data set of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set mag_track_solution
#is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

#Smooth the completed tracking of the 405 kyr eccentricity cycle in the wavelet spectra

mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)




mag_405_ecc  <- extract_signal(
tracked_cycle_curve = mag_track_complete,
wavelet = mag_wt,
period_up = 1.2,
period_down = 0.8,
add_mean = TRUE,
tracked_cycle_period = 405,
extract_cycle = 405,
tune = FALSE,
plot_residual = FALSE
)

#extract the amplitude  of the 405 kyr eccentricity cycle
mag_ampl <- extract_amplitude(
signal = mag_405_ecc,
```

```
pts=3,
genplot = FALSE,
ver_results = FALSE,
keep_editable=FALSE)
```

extract_power *Extract power from a wavelet spectra*

Description

Extracts the spectral power from a wavelet spectra in the depth domain using a traced period and boundaries surround the traced period. The extraction of spectral is useful for cyclostratigraphic studies because the spectral power of an astronomical cycle is modulated by higher order astronomical cycles. The spectral power record from an astronomical cycle can thus be used as a proxy for amplitude modulating cycles The traced period result from the `track_period_wavelet` function with boundaries is used to extract spectral power in the depth domain from a wavelet spectra.

Usage

```
extract_power(
  completed_series = NULL,
  wavelet = NULL,
  period_up = 1.2,
  period_down = 0.8,
  tracked_cycle_period = NULL,
  extract_cycle_power = NULL
)
```

Arguments

`completed_series`
: Traced period result from the `track_period_wavelet` function completed using the `completed_series`. The input can be pre-smoothed using the the `loess_auto` function.

`wavelet`
: Wavelet object created using the `analyze_wavelet` function.

`period_up`
: Upper period as a factor of the to be extracted power `Default=1.2`.

`period_down`
: Lower period as a factor of the to be extracted power `Default=0.8`.

`tracked_cycle_period`
: Period of the tracked cycle (make sure that `tracked_cycle_period`) and `extract_cycle_power`) are of the same unit (either depth or time domain).

`extract_cycle_power`
: Period of the cycle for which the power will be extracted (make sure that `extract_cycle_power`) and `tracked_cycle_period`) are of the same unit (either depth or time domain).

Value

Returns a matrix with 3 columns. The first column is depth/time. The second column is extracted power. The third column is extracted power/total power.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo. The assignment of the standard deviation of the uncertainty of the wavelet is based on the work of Gabor (1946) and Russell et al., (2016) The functionality of this function is is inspired by the integratePower function of the 'astrochron' R package.

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#Extract the power of the 405 kyr eccentricity cycle from the the magnetic
# susceptibility data set of De pas et al., (2018)
#Perform the CWT on the magnetic susceptibility data set of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set mag_track_solution
#is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
```

```
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

#Smooth the completed tracking of the 405 kyr eccentricity cycle in the wavelet spectra

mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)

#extract the spectral power of the 405 kyr eccentricity cycle
mag_power <- extract_power(
completed_series = mag_track_complete,
wavelet = mag_wt,
period_up = 1.2,
period_down = 0.8,
tracked_cycle_period = 405,
extract_cycle_power = 405
)
```

`extract_power_stable` *Extract power from a wavelet spectra by using a constant period/duration*

Description

Extract spectral power from the wavelet using a constant period/duration and boundaries as selection criteria. The extraction of spectral is useful for cyclostratigraphic studies because the spectral power of an astronomical cycle is modulated by higher order astronomical cycles. The spectral power record from an astronomical cycle can thus be used as a proxy for amplitude modulating cycles. The spectral power is extracted from a wavelet spectra which was created using the `analyze_wavelet` function for a given, `cycle`, `period_up` and `period_down`

Usage

```
extract_power_stable(
  wavelet = NULL,
  cycle = NULL,
  period_up = 1.2,
  period_down = 0.8
)
```

Arguments

`wavelet`	Wavelet object created using the `analyze_wavelet` function.

`cycle`	Period of cycle for which the power will be extracted from the record.
`period_up`	Species the upper period of the to be extracted power `Default=1.2`.
`period_down`	specifies the lower period of the to be extracted power `Default=0.8`.

Value

Returns a matrix with 3 columns. The first column is depth/time. The second column is extracted power. The third column is extracted power/total power.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998). The functionality of this function is is inspired by the integratePower function of the 'astrochron' R package

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#Extract the spectral power of the 210 yr de Vries cycle from the Total Solar
#Irradiance data set of Steinhilber et al., (2012).

TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )
TSI_wt_pwr_de_Vries_cycle <-  extract_power_stable(
 wavelet = TSI_wt,
 cycle = 210,
 period_up = 1.2,
 period_down = 0.8
)
```

extract_signal *Extract signal from a wavelet spectra using a traced period curve*

Description

Extract signal power from the wavelet in the depth domain using the traced period.

Usage

```
extract_signal(
  tracked_cycle_curve = NULL,
  wavelet = NULL,
  period_up = 1.2,
  period_down = 0.8,
  add_mean = TRUE,
  tracked_cycle_period = NULL,
  extract_cycle = NULL,
  tune = FALSE,
  plot_residual = FALSE
)
```

Arguments

- `tracked_cycle_curve` Traced period result from the `track_period_wavelet` function completed using the `completed_series`. The input can be pre-smoothed using the the `loess_auto` function.
- `wavelet` wavelet object created using the `analyze_wavelet` function.
- `period_up` Upper period as a factor of the to be extracted cycle `Default=1.2`.
- `period_down` Lower period as a factor of the to be extracted cycle `Default=0.8`.
- `add_mean` Add mean to the extracted cycle `Default=TRUE`.
- `tracked_cycle_period` Period in time of the traced cycle.
- `extract_cycle` Period of the to be extracted cycle.
- `tune` Convert record from the depth to the time domain using the traced period `Default=FALSE`.
- `plot_residual` Plot the residual signal after extraction of set cycle `Default=FALSE`.

Value

Returns a matrix with 2 columns The first column is depth/time The second column is extracted signal

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Examples

```
#Extract the 405 kyr eccentricity cycle from the the magnetic susceptibility \cr
#record of the Sullivan core and use the Gabor uncertainty principle to define \cr
#the mathematical uncertainty of the analysis and use a factor of that standard \cr
#deviation to define boundaries.

#Perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                    wavelet=mag_wt,
#                                    n.levels = 100,
#                                    periodlab = "Period (metres)",
#                                    x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution


mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)

# extract the 405 kyr eccentricity cycle from the wavelet spectrum and use the \cr
# tracked cycle curve and set factors of the extracted cycle as boundaries
```

```
mag_405_ecc  <- extract_signal(
tracked_cycle_curve = mag_track_complete,
wavelet = mag_wt,
period_up = 1.2,
period_down = 0.8,
add_mean = TRUE,
tracked_cycle_period = 405,
extract_cycle = 405,
tune = FALSE,
plot_residual = FALSE
)
```

`extract_signal_stable` *Extract a signal/cycle from a wavelet spectra using a set period and boundaries*

Description

Extracts a cycle from the wavelet object created using the `analyze_wavelet` function using a fixed period and fixed period boundaries defined as factors of the original cycle

Usage

```
extract_signal_stable(
  wavelet = NULL,
  cycle = NULL,
  period_up = 1.2,
  period_down = 0.8,
  add_mean = TRUE,
  plot_residual = FALSE,
  keep_editable = FALSE
)
```

Arguments

`wavelet`	Wavelet object created using the `analyze_wavelet` function.
`cycle`	Period of the cycle which needs to be extracted.
`period_up`	Specifies the upper period as a factor of the to be extracted cycle `Default=1.2`.
`period_down`	Specifies the lower period as a factor of the to be extracted cycle `Default=0.8`.
`add_mean`	Add mean to the extracted cycle `Default=TRUE`.
`plot_residual`	plot the residual signal after extraction of set cycle `Default=FALSE`.
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`

Value

#'Returns a matrix with 2 columns. The first column is time/depth. The second column is the extracted signal/cycle.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar
#Irradiance data set of Steinhilber et al., (2012)

#Perform the CWT
TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

#Extract the 210 yr de Vries cycle from the wavelet spectra
de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE,
keep_editable=FALSE)
```

extract_signal_stable_V2 *Extract signal from a wavelet spectrum using a upper and lower period boundary*

Description

Extract a signal from the wavelet using a upper and lower period boundary

Usage

```
extract_signal_stable_V2(
  wavelet = NULL,
  period_max = NULL,
  period_min = NULL,
  add_mean = TRUE,
  plot_residual = FALSE,
  keep_editable = FALSE
)
```

Arguments

wavelet	wavelet object created using the analyze_wavelet function.
period_max	Maximum period (upper boundary) to be used to extract a cycle.
period_min	Minimum period (lower boundary) to be used to extract a cycle.
add_mean	Add mean to the extracted cycle Default=TRUE.
plot_residual	Plot the signal from which the extracted cycle is subtracted Default=FALSE.
keep_editable	Keep option to add extra features after plotting Default=FALSE

Value

Signal extracted from the wavelet spectra. Output is a matrix with the first column being depth/time and the second column is the cycle extracted from the proxy record.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar
# Irradiance data set of Steinhilber et al., (2012)

TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

de_Vries_cycle <- extract_signal_stable_V2(wavelet=TSI_wt,
period_max = 240,
period_min = 180,
add_mean=TRUE,
plot_residual=FALSE,
keep_editable=FALSE)
```

`extract_signal_standard_deviation`

Extract a signal using standard deviation

Description

Extract signal from a wavelet spectra in the depth domain using a the standard deviation of the omega (number of cycles) as boundaries. The uncertainty is based on the Gabor uncertainty principle applied to the continuous wavelet transform using a Morlet wavelet. The calculated uncertainty is the underlying analytical uncertainty which is the result of applying the Gabor uncertainty principle to the continuous wavelet transform using a Morlet wavelet.

Usage

```
extract_signal_standard_deviation(
  wavelet = NULL,
  tracked_cycle_curve = NULL,
  multi = 1,
  extract_cycle = NULL,
  tracked_cycle_period = NULL,
  add_mean = TRUE,
```

```
  tune = FALSE,
  genplot_uncertainty_wt = FALSE,
  genplot_extracted = FALSE,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices"
)
```

Arguments

`wavelet` Wavelet object created using the `analyze_wavelet` function.

`tracked_cycle_curve` Curve of the cycle tracked using the `track_period_wavelet` function. Any input (matrix or data frame) in which the first column is depth or time and the second column is period should work.

`multi` multiple of the standard deviation to be used as boundaries for the cycle extraction `Default=1`.

`extract_cycle` Period of the cycle to be extracted.

`tracked_cycle_period` Period of the tracked cycle.

`add_mean` Add mean to the extracted cycle `Default=TRUE`.

`tune` Tune data set using the `Default=tracked_cycle_curve` curve `Default=FALSE`.

`genplot_uncertainty_wt` Generate a wavelet spectra plot with the tracked curve and its analytical uncertainty based the Gabor uncertainty principle applied continuous wavelet transform using a Morlet wavelet on superimposed on top of it. In the plot the red curve and blue curves are the upper and lower bounds based on the `multi` parameter which x-times the standard deviation of uncertainty. The black curve is the `Default=FALSE` curve.

`genplot_extracted` Generates a plot with the data set and the extracted cycle on top `Default=FALSE` of it.

`keep_editable` Keep option to add extra features after plotting `Default=FALSE`

`palette_name` Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

color_brewer
: Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "`Default=grDevices`

Value

Signal extracted from the wavelet spectra. Output is a matrix with the first column being depth/time and the second column is the astronomical cycle extracted from the proxy record

If `genplot_uncertainty_wt=TRUE` then a wavelet spectra will be plotted with the uncertainty superimposed on top of it. In the plot the red curve and blue curves are the upper and lower bounds based on the `multi` parameter.The black curve is the `Default=tracked_cycle_curve` curve. If `genplot_extracted=TRUE` plot with the data set and the extracted cycle on top of it will be plotted.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998). The assignment of the standard deviation of the uncertainty of the wavelet is based on the work of Gabor (1946) and Russell et al., (2016)

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Gabor, Dennis. "Theory of communication. Part 1: The analysis of information." Journal of the Institution of Electrical Engineers-part III: radio and communication engineering 93, no. 26 (1946): 429-441.http://genesis.eecg.toronto.edu/gabor1946.pdf

Russell, Brian, and Jiajun Han. "Jean Morlet and the continuous wavelet transform. " CREWES Res. Rep 28 (2016): 115. https://www.crewes.org/Documents/ResearchReports/2016/CRR201668.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#Extract the 405 kyr eccentricity cycle from the magnetic susceptibility
#record of the Sullivan core of Pas et al., (2018) and use the Gabor
# uncertainty principle to define the mathematical uncertainty of the
# analysis and use a factor of that standard deviation to define
# boundaries

# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                    wavelet=mag_wt,
#                                    n.levels = 100,
#                                    periodlab = "Period (metres)",
#                                    x_lab = "depth (metres)",
#                                    palette_name="rainbow",
#                                    color_brewer="grDevices")

#Instead of tracking, the tracked solution data set mag_track_solution is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)

# extract the 405 kyr eccentricity cycle from the wavelet spectrum and use
# the Gabor uncertainty principle to define the mathematical uncertainty of
# the analysis and use a multiple of the derived standard deviation to define boundaries

mag_405_ecc <- extract_signal_standard_deviation(
wavelet = mag_wt,
tracked_cycle_curve = mag_track_complete,
multi = 1,
extract_cycle = 405,
tracked_cycle_period = 405,
add_mean = TRUE,
```

```
tune = FALSE,
genplot_uncertainty_wt = FALSE,
genplot_extracted = FALSE,
keep_editable=FALSE,
palette_name="rainbow",
color_brewer="grDevices"
)
```

---

flmw — *Fit linear models to spectral peaks extracted from the wavelet spectra to astronomical cycles multiplied by sedimentation rate x*

Description

The flmw function is used calculate the linear correlation for a list of astronomical cycles transformed using a range of sedimentation rates and then compared to spectral peaks of a wavelet spectra

Usage

```
flmw(
  wavelet = NULL,
  sedrate_low = NULL,
  sedrate_high = NULL,
  spacing = NULL,
  cycles = c(NULL),
  x_lab = "depth",
  y_lab = "sedrate",
  run_random = FALSE,
  rand_simulations = 1000,
  run_multicore = FALSE,
  genplot = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_res = 2,
  keep_editable = FALSE,
  verbose = FALSE
)
```

Arguments

wavelet	Wavelet object created using the analyze_wavelet function
sedrate_low	Minimum sedimentation rate (cm/kyr)for which the sum of maximum spectral power is calculated for.
sedrate_high	Maximum sedimentation rate (cm/kyr) for which the sum of maximum spectral power is calculated for.

spacing	Spacing (cm/kyr) between sedimentation rates
cycles	Astronomical cycles (in kyr) for which the combined sum of maximum spectral power is calculated for
x_lab	label for the y-axis `Default="depth"`
y_lab	label for the y-axis `Default="sedrate"`
run_random	run multiple simulation to calculate percentile against the 0 hypothesis
rand_simulations	nr of simulations to calculate percentile against the 0 hypothesis
run_multicore	run simulation using multiple cores `Default=FALSE` the simulation is run at x-2 cores to allow the 2 remaining processes to run background processes
genplot	Generate plot `Default="FALSE"`
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. `"Default=grDevices`
plot_res	options 1-8 option 1: slope coefficient, option 2: r squared, option 3: nr of components, option 4: difference to the origin , option 5: slope coefficient percentile option 6: r squared percentile, option 7: nr of components percentile, option 8: difference to the origin percentile `Default=2`
keep_editable	Keep option to add extra features after plotting `Default=FALSE`
verbose	Print text `Default=FALSE`.

Value

Returns a list which contains 10 elements element 1: slope coefficient element 2: r squared element 3: nr of components element 4: difference to the origin element 5: slope coefficient percentile element 6: r squared percentile element 7: nr of components percentile, element 8: difference to the origin percentile element 9: y-axis values of the matrices which is sedimentation rate element 10: x-axis values of the matrices which is depth

Author(s)

Based on the eAsm function of the 'astrochron' R package and the 'eCOCO' and 'COCO' function of the 'Acycle' software

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Acycle: Time-series analysis software for paleoclimate research and education, Mingsong Li, Linda Hinnov, Lee Kump, Computers & Geosciences,Volume 127,2019,Pages 12-22,ISSN 0098-3004, <doi:10.1016/j.cageo.2019.02.011>

Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing, Mingsong Li, Lee R. Kump, Linda A. Hinnov, Michael E. Mann, Earth and Planetary Science Letters,Volume 501, T2018,Pages 165-179,ISSN 0012-821X,<doi:10.1016/j.epsl.2018.08.041>

Examples

```
#estimate sedimentation rate for the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018).

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

sedrates <- flmw(wavelet = mag_wt,
    sedrate_low = 0.5,
    sedrate_high = 4,
    spacing = 0.05,
    cycles = c(2376,1600,1180,696,406,110),
    x_lab = "depth",
    y_lab = "sedrate",
    run_random = FALSE,
    rand_simulations = 50, # increase to get better constrainted resutls
    run_multicore = FALSE,
    genplot = FALSE,
    palette_name = "rainbow",
    color_brewer = "grDevices",
    plot_res = 2,
    keep_editable=FALSE,
    verbose=FALSE)
```

geo_col *Generate standard color codes for the Geological Time Scale*

Description

Generates the R color code which corresponds its respective geological subdivision

Usage

```
geo_col(name = NULL)
```

Arguments

name	Name of the geologchronological subdivision

Value

Returns the color code of the geological subdivision

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

Examples

```
#generate the Silurian part of the GTS
plot.new()
plot(
 x = c(0, 1),
 y = c(419.2, 443.8),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(419, 444))
)            # Draw empty plot

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Rhuddanian"),
 col =geo_col("Rhuddanian")
)

text(
 0.85,geo_mid("Rhuddanian"),
 "Rhuddanian",
```

```
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Aeronian"),
 col =geo_col("Aeronian")
)

text(
 0.85,geo_mid("Aeronian"),
 "Aeronian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Telychian"),
 col =geo_col("Telychian")
)

text(
 0.85,geo_mid("Telychian"),
 "Telychian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Sheinwoodian"),
 col =geo_col("Sheinwoodian")
)

text(
 0.85,geo_mid("Sheinwoodian"),
 "Sheinwoodian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Homerian"),
 col =geo_col("Homerian")
)
```

```
text(
 0.85,geo_mid("Homerian"),
 "Homerian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Gorstian"),
 col =geo_col("Gorstian")
)

text(
 0.85,geo_mid("Gorstian"),
 "Gorstian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Ludfordian"),
 col =geo_col("Ludfordian")
)

text(
 0.85,geo_mid("Ludfordian"),
 "Ludfordian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Pridoli_Age"),
 col =geo_col("Pridoli_Age")
)



polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Pridoli"),
 col =geo_col("Pridoli")
)

text(
```

```
 0.5,geo_mid("Pridoli"),
 "Pridoli",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Ludlow"),
 col =geo_col("Ludlow")
)

text(
 0.5,geo_mid("Ludlow"),
 "Ludlow",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Wenlock"),
 col =geo_col("Wenlock")
)

text(
 0.5,geo_mid("Wenlock"),
 "Wenlock",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Llandovery"),
 col =geo_col("Llandovery")
)

text(
 0.5,geo_mid("Llandovery"),
 "Llandovery",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0, 0.33, 0.33, 0),
 y = geo_loc("Silurian"),
```

```
 col =geo_col("Silurian")
)

text(
 0.165,geo_mid("Silurian"),
 "Silurian",
 cex = 1,
 col = "black",
 srt = 0
)
```

---

geo_loc *Generates ages for the boundaries of a geochronological subdivision*

Description

Generates ages for the boundaries of a geochronological subdivision which is based on the Geological Time Scale

Usage

```
geo_loc(name = NULL)
```

Arguments

name	Name of the geologchronological subdivision

Value

Returns the ages of the boundary of a geochronological subdivision which can then be added to a polygon object

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

Examples

```
#generate the Silurian part of the GTS
plot.new()
plot(
 x = c(0, 1),
 y = c(419.2, 443.8),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
```

```
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(419, 444))
)            # Draw empty plot

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Rhuddanian"),
 col =geo_col("Rhuddanian")
)

text(
 0.85,geo_mid("Rhuddanian"),
 "Rhuddanian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Aeronian"),
 col =geo_col("Aeronian")
)

text(
 0.85,geo_mid("Aeronian"),
 "Aeronian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Telychian"),
 col =geo_col("Telychian")
)

text(
 0.85,geo_mid("Telychian"),
 "Telychian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Sheinwoodian"),
 col =geo_col("Sheinwoodian")
)
```

```
text(
 0.85,geo_mid("Sheinwoodian"),
 "Sheinwoodian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Homerian"),
 col =geo_col("Homerian")
)

text(
 0.85,geo_mid("Homerian"),
 "Homerian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Gorstian"),
 col =geo_col("Gorstian")
)

text(
 0.85,geo_mid("Gorstian"),
 "Gorstian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Ludfordian"),
 col =geo_col("Ludfordian")
)

text(
 0.85,geo_mid("Ludfordian"),
 "Ludfordian",
 cex = 1,
 col = "black",
 srt = 0
)
```

```
polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Pridoli_Age"),
 col =geo_col("Pridoli_Age")
)



polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Pridoli"),
 col =geo_col("Pridoli")
)

text(
 0.5,geo_mid("Pridoli"),
 "Pridoli",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Ludlow"),
 col =geo_col("Ludlow")
)

text(
 0.5,geo_mid("Ludlow"),
 "Ludlow",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Wenlock"),
 col =geo_col("Wenlock")
)

text(
 0.5,geo_mid("Wenlock"),
 "Wenlock",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
```

```
 y = geo_loc("Llandovery"),
 col =geo_col("Llandovery")
)

text(
 0.5,geo_mid("Llandovery"),
 "Llandovery",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0, 0.33, 0.33, 0),
 y = geo_loc("Silurian"),
 col =geo_col("Silurian")
)

text(
 0.165,geo_mid("Silurian"),
 "Silurian",
 cex = 1,
 col = "black",
 srt = 0
)
```

---

geo_mid *Generate the mean age of a geological subdivision*

Description

Generates the mean age of a geological subdivision which is based on the Geological Time Scale

Usage

```
geo_mid(name = NULL)
```

Arguments

name	Name of the geologchronological subdivision

Value

Returns the mean age of the geochronological subdivision

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

Examples

```
#generate the Silurian part of the GTS
plot.new()
plot(
 x = c(0, 1),
 y = c(419.2, 443.8),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(419, 444))
)            # Draw empty plot

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Rhuddanian"),
 col =geo_col("Rhuddanian")
)

text(
 0.85,geo_mid("Rhuddanian"),
 "Rhuddanian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Aeronian"),
 col =geo_col("Aeronian")
)

text(
 0.85,geo_mid("Aeronian"),
 "Aeronian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Telychian"),
 col =geo_col("Telychian")
)

text(
 0.85,geo_mid("Telychian"),
 "Telychian",
```

```
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Sheinwoodian"),
 col =geo_col("Sheinwoodian")
)

text(
 0.85,geo_mid("Sheinwoodian"),
 "Sheinwoodian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Homerian"),
 col =geo_col("Homerian")
)

text(
 0.85,geo_mid("Homerian"),
 "Homerian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Gorstian"),
 col =geo_col("Gorstian")
)

text(
 0.85,geo_mid("Gorstian"),
 "Gorstian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Ludfordian"),
 col =geo_col("Ludfordian")
```

```
)

text(
 0.85,geo_mid("Ludfordian"),
 "Ludfordian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Pridoli_Age"),
 col =geo_col("Pridoli_Age")
)



polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Pridoli"),
 col =geo_col("Pridoli")
)

text(
 0.5,geo_mid("Pridoli"),
 "Pridoli",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Ludlow"),
 col =geo_col("Ludlow")
)

text(
 0.5,geo_mid("Ludlow"),
 "Ludlow",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Wenlock"),
 col =geo_col("Wenlock")
)
```

```
text(
 0.5,geo_mid("Wenlock"),
 "Wenlock",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Llandovery"),
 col =geo_col("Llandovery")
)

text(
 0.5,geo_mid("Llandovery"),
 "Llandovery",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0, 0.33, 0.33, 0),
 y = geo_loc("Silurian"),
 col =geo_col("Silurian")
)

text(
 0.165,geo_mid("Silurian"),
 "Silurian",
 cex = 1,
 col = "black",
 srt = 0
)
```

---

grey *Grey scale record IODP 926 of Zeeden et al., (2013)*

Description

IODP 926 grey scale record of Zeeden et al., (2013) for the (154-174m) interval. The (154-174m) interval spans the Miocene.

Details

Column 1: depth (meters)
Column 2: greyscale value

References

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

`grey_track` *Tracking points of the precession (22 kyr cycle) IODP 926 grey scale (154-174m) record of Zeeden et al., (2013)*

Description

Example data which consists of tracking points of the precession (22 kyr cycle) in the wavelet scalogram of the IODP 926 grey scale (154-174m) record of Zeeden et al., (2013)

Details

Column 1: Depth (meters)
Column 2: period (meters)

References

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

`GTS_info` *Information of the Geological timescale 2020*

Description

GTS_info data set consists the information of the Geological timescale 2020 including the color data of Ogg et al., (2021) The ages, durations, uncertainties and colors of the Geological timescale 2020 are included in the data set

Details

Column 1: name
Column 2: type
Column 1: top age
Column 1: top error
Column 1: bottom age
Column 1: bottom error

Column 1: Cyan value
Column 1: Magenta value
Column 1: Yellow value
Column 1: Key value
Column 1: Red Value
Column 1: Green value
Column 1: Blue value
Column 1: font style
Column 1: font color

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

Hilbert_transform *Perform a Hilbert transform on a signal*

Description

Extract the amplitude modulation using the Hilbert transform.

Usage

```
Hilbert_transform(data = NULL, demean = TRUE, nr_pad = 100)
```

Arguments

data	Input is a time series with the first column being depth or time and the second column being a proxy.
demean	Remove the mean from the time series.
nr_pad	nr of points added tot the top and bottom of the data set to mitigate the edging effect of the Hilbert transform.

Value

Returns a matrix with 2 columns. The first column is depth/time. The second column is the Hilbert transform of the signal.

Author(s)

Based on the the inst.pulse function of the 'DecomposeR' R package.

References

Wouters, S., Crucifix, M., Sinnesael, M., Da Silva, A.C., Zeeden, C., Zivanovic, M., Boulvain, F., Devleeschouwer, X., 2022, "A decomposition approach to cyclostratigraphic signal processing". Earth-Science Reviews 225 (103894). <doi:10.1016/j.earscirev.2021.103894>

Huang, Norden E., Zhaohua Wu, Steven R. Long, Kenneth C. Arnold, Xianyao Chen, and Karin Blank. 2009. "On Instantaneous Frequency". Advances in Adaptive Data Analysis 01 (02): 177–229. <doi:10.1142/S1793536909000096>

Examples

```
#Example in which the Hilbert transform (eg. amplitude modulation) of the ~210yr
#de Vries cycle is extracted from the Total Solar Irradiance data set of
#Steinhilber et al., (2012)

#Perform the CWT
TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

#Extract the 210 yr de Vries cycle from the wavelet spectra
de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE)

#Perform the Hilbert transform on the amplitude record of the 210 yr de Vries
# cycle which was extracted from the wavelet spectra

de_Vries_cycle_hilbert <- Hilbert_transform(data=de_Vries_cycle,demean=TRUE)
```

---

lag_1 *lag-1 autocorrelation coefficient*

Description

The lag_1 function calculates the lag-1 autocorrelation coefficient using a windowed analysis monte carlo analysis

Usage

```
lag_1(
  data = NULL,
  n_sim = 10,
  run_multicore = FALSE,
  win_max = NULL,
  win_min = NULL,
  verbose = FALSE
)
```

Arguments

data	Input data set should consist of a matrix with 2 columns with first column being depth and the second column being a proxy
n_sim	number of simulations to be ran
run_multicore	Run function using multiple cores Default="FALSE"
win_max	maximum window size
win_min	minimum window size
verbose	print text

Value

Returns a matrix which contains 3 columns column 1: depth/time matrix column 2: mean autocorrelation coefficient column 3: sd autocorrelation coefficient

Author(s)

Michiel Arts

Examples

```
#The example uses the magnetic susceptibility data set of Pas et al., (2018).
# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (meters)",
#                                   x_lab = "depth (meters)")

#Instead of tracking, the tracked solution data set mag_track_solution is used
```

```
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE)
#convert period in meters to sedrate depth vs time
mag_track_time<- curve2tune(data=mag,
                           tracked_cycle_curve=mag_track_complete,
                           tracked_cycle_period=405,
                           genplot = FALSE,
                           keep_editable=FALSE)

mag_lag_1 <- lag_1(data = mag_track_time,n_sim = 10,
run_multicore = FALSE,
win_max = 505,
win_min = 150,
verbose=FALSE)
```

`lithlog_disc` *Discriticizes lithologs*

Description

Discriticizes lithologs to allow further time-series analysis first the Greatest common divisor/highest common factor is calculated which is then used to discriticize the litholog to an evenly sampled data series. The function is designed to place the boundary at the original depth level of the bed boundaries. The Greatest common divisor/highest common factor can be a very small number as such the discriticized data set can be large which impacts computational performance later on therefore a linear interpolation option is added to downscale the data to allow for computational efficiency later on. This is made to discriticize lithologs created using the 'StratigrapheR' package. as such the same data format for input is used. eg. column 1 is bottom of the bed, column 2 is top of bed, column is depth rank/proxy value

Usage

```
lithlog_disc(
  litholog = NULL,
  subset_fact = 10,
```

```
  lin_interp = FALSE,
  dt = NULL,
  genplot = FALSE,
  x_lab = "rank",
  y_lab = "depth (m)",
  keep_editable = FALSE
)
```

Arguments

`litholog`	litholog input matrix with 3 columns column 1 is bottom of the bed, column 2 is top of bed, column is depth rank/proxy value
`subset_fact`	subset factor which is x times the greatest common divider `Default=10`.
`lin_interp`	Linear interpolation of the data set `Default=FALSE`
`dt`	step size `Default=NULL`.
`genplot`	generate plot `Default=FALSE`
`x_lab`	label for the y-axis `Default="rank"`
`y_lab`	label for the y-axis `Default="depth (m)"`
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`

Value

Returns a matrix with 2 columns, the first column is depth the second columns is the depth/rank proxy If genplot is `Default=TRUE` then a plot of the discriticizes time series is plotted

References

Wouters, S., Da Silva, A.-C., Boulvain, F., and Devleeschouwer, X.. 2021. StratigrapheR: Concepts for Litholog Generation in R. The R Journal. <doi:10.32614/RJ-2021-039>

Examples

```
# Convert depth rank record to a discrete proxy record to allow for further
# analysis in which discrete time series are needed
depth_rank_example_disc <- lithlog_disc(litholog = depth_rank_example,
           subset_fact = 10,
           genplot = FALSE,
           x_lab = "rank",
           y_lab = "depth (m)",
           keep_editable=FALSE)
```

loess_auto — *Perform an automatically loess based smoothing of a time series*

Description

Perform an automatically loess based smoothing of a time series. The local polynomial regression with automatic smoothing parameter selection is based on an optimization using the 'aicc' bias-corrected 'AIC' criterion and the 'gcv' generalized cross-validation criterion.

Usage

```
loess_auto(
  time_series = NULL,
  genplot = FALSE,
  print_span = FALSE,
  keep_editable = FALSE
)
```

Arguments

time_series	Input is a time series with the first column being depth or time and the second column being a proxy
genplot	Option to generate plot `Default=TRUE`. The plot will consist of the original signal in blue, the smoothed plot is displayed in black and the + and - 1 sd bounds of the smoothing are displayed in red.
print_span	Print span length as a fraction of the total length of the record.
keep_editable	Keep option to add extra features after plotting `Default=FALSE`

Value

A matrix with 3 columns. The first column is depth/time. The second column is the smoothed curve. The third column is difference between the original curve and the smoothed curve.

Author(s)

Based on the the loess.as function of the 'fANCOVA' R package.

References

Cleveland, W. S. (1979) Robust locally weighted regression and smoothing scatter plots. Journal of the American Statistical Association. 74, 829–836. <doi:10.1080/01621459.1979.10481038> Hurvich, C.M., Simonoff, J.S., and Tsai, C.L. (1998), Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion. Journal of the Royal Statistical Society B. 60, 271–293 <doi:10.1111/1467-9868.00125> Golub, G., Heath, M. and Wahba, G. (1979). Generalized cross validation as a method for choosing a good ridge parameter. Technometrics. 21, 215–224. <doi:10.2307/1268518>

Examples

```
#'smooth the period curve of the 405 kyr eccentricity cycle extracted from
# the magnetic susceptibility data set of Pas et al., (2018)
#perform the CWT on the magnetic susceptibility data set of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set mag_track_solution is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE,
  keep_editable=FALSE
)

#Smooth the completed tracking of the 405 kyr eccentricity cycle as tracked in the wavelet spectra
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = FALSE, print_span = FALSE,keep_editable=FALSE)
```

---

mag *Magnetic susceptibility data of the Sullivan core of Pas et al., (2018)*

Description

The magnetic susceptibility data set consists of the magnetic susceptibility measurements of Pas et al., (2018), which measured the magnetic susceptibility on the Sullivan core which is of Famennian age.

Details

Column 1: depth value (meters depoth)
Column 2: magnetic susceptibility value

References

Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu, Cyclostratigraphic calibration of the Famennian stage (Late Devonian, Illinois Basin, USA), Earth and Planetary Science Letters, Volume 488, 2018, Pages 102-114, ISSN 0012-821X, <doi:1016/j.epsl.2018.02.010>

`mag_track_solution` *Period of the 405 kyr ecc cycle in the magnetic susceptibility record of the Sullivan core*

Description

Data points which give the period (in meters) of the 405 kyr eccentricity cycle tracked in the wavelet scalogram of the magnetic susceptibility record of the Sullivan core
The period was tracked using the `track_period_wavelet` function
The tracking is based on the original age model of Pas et al., (2018)

Details

Column 1: Depth (meters)
Column 2: tracked period of 405 kyr eccentricity cycle (meters)

References

Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu, Cyclostratigraphic calibration of the Famennian stage (Late Devonian, Illinois Basin, USA), Earth and Planetary Science Letters, Volume 488, 2018, Pages 102-114, ISSN 0012-821X, <doi:10.1016/j.epsl.2018.02.010>

`max_detect` *Detect and filter out all maxima in a signal*

Description

The `max_detect` function is used to detect and filter out local maxima in a sinusoidal signal.

Usage

```
max_detect(data = NULL, pts)
```

Arguments

data
: Matrix or data frame with the first column being depth or time and the second column being a proxy

pts
: The pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=3`

Value

#Returns a matrix with 2 columns first column is depth/time the second column are local maxima values

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar
#Irradiance data set of Steinhilber et al., (2012)
#after which all maxima are extracted

TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE)


min_de_Vries_cycle <- min_detect(de_Vries_cycle)
```

---

minimal_tuning *Create an age model using minimal tuning*

Description

Create an age model using the minimal tuning technique. This means that the distance between 2 peaks of an extracted cycle are set to duration of the interpreted astronomical cycle

Usage

```
minimal_tuning(
  data = NULL,
  pts = 5,
  cycle = 405,
  tune_opt = "max",
  output = 0,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

`data`	Input is an cycle extracted filtered in the depth domain
`pts`	The pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=5`
`cycle`	duration in kyr of the filtered/extracted cycle
`tune_opt`	tuning options "min", "max" and "minmax" use minima, maxima or both of the cyclic signal to create the age model `Default="max"`
`output`	#'The output depends on the output setting If output = 0 output is a matrix of with 4 columns being; depth,proxy,sedimentation rate and time If output = 1 output is a matrix of with 2 columns being; depth and sedimentation rate #'If output = 2 output is a matrix of with 2 columns being; depth and time
`genplot`	Keep option to add extra features after plotting `Default=FALSE`
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`

Value

The output depends on the output setting If output = 0 output is a matrix of with 4 columns being (depth,proxy,sedimentation rate and time) If genplot = TRUE 4 plots are generated; depth vs proxy, depth vs sedimentation rate, depth vs time and time vs proxy If output = 1 output is a matrix of with 2 columns being (depth and sedimentation rate) If genplot = TRUE a plot of depth vs sedimentation rate is generated If output = 2 output is a matrix of with 2 columns being (depth and time) If genplot = TRUE a plot of depth vs time is generated

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
# Extract the 405kyr eccentricity cycle from the wavelet scalogram
# from the magnetic susceptibility record f the Sullivan core
# of Pas et al., (2018) and then create a age model using minimal tuning
# (e.g.) set the distance between peaks to 405 kyr

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)


mag_405 <- extract_signal_stable_V2(
 wavelet = mag_wt,
 period_max = 4,
 period_min = 2,
 add_mean = FALSE,
 plot_residual = FALSE,
 keep_editable = FALSE
)

mag_405_min_tuning <- minimal_tuning(data = mag_405,
pts = 5,
cycle = 405,
tune_opt = "max",
output = 0,
genplot = FALSE,
keep_editable = FALSE)
```

---

min_detect — *Detect and filter out all minima in a signal*

Description

The min_detect function is used to detect and filter out local minima in a sinusoidal signal

Usage

```
min_detect(data = NULL, pts = 3)
```

Arguments

`data` Matrix or data frame with first column being depth or time and the second column being a proxy

`pts` the pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=3`

Value

#Returns a matrix with 2 columns first column is depth/time the second column are local minima values

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar
#Irradiance data set of Steinhilber et al., (2012)
#after which all minima are extracted

TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE)


min_de_Vries_cycle <- min_detect(de_Vries_cycle)
```

`model_red_noise_wt` *Models average spectral power based curves based on a red-noise signal generated using the characteristics of an input signal.*

Description

The model_red_noise_wt function is used to generate average spectral power curves based on and input signal and set wavelet settings.

Usage

```
model_red_noise_wt(
  wavelet = NULL,
  n_simulations = NULL,
  run_multicore = FALSE,
  verbose = FALSE
)
```

Arguments

wavelet	Wavelet object created using the analyze_wavelet function.
n_simulations	Number of red noise simulations.
run_multicore	run simulation using multiple cores Default=FALSE the simulation is run at x-2 cores to allow the 2 remaining processes to run background processes.
verbose	Print text Default=FALSE.

Value

Returns a matrix in which each column represents the average spectral power resulting from a red-noise run.

Author(s)

Code based on the "analyze.wavelet" function of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#'#generate average spectral power curves based on red noise curves which are
# based on the magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#increase n_simulations to better define the red noise spectral power curve
mag_wt_red_noise <- model_red_noise_wt(wavelet=mag_wt,
n_simulations=10, # increase number for better constrained results
run_multicore=FALSE,
verbose=FALSE)
```

---

percentile_from_red_noise

Calculate average spectral power from red noise curves for a given percentile

---

Description

The `percentile_from_red_noise` function is used to generate and average spectral power curve based on a set percentile based. To generate the percentile curve the results of the `model_red_noise_wt` function are used.

Usage

```
percentile_from_red_noise(red_noise = NULL, wavelet = NULL, percentile = NULL)
```

Arguments

`red_noise`	Red noise curves generated using the `model_red_noise_wt` function.
`wavelet`	Wavelet object created using the `analyze_wavelet` function.
`percentile`	Percentile value (0-1).

Value

Returns a matrix with 2 columns.
The first column is the period (m).
The second column is the spectral power at percentile x based on
the red noise modelling runs.

Examples

```
#'#generate average spectral power curves based on red noise curves which are
# based on the magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#increase n_simulations to better define the red noise spectral power curve
mag_wt_red_noise <- model_red_noise_wt(wavelet=mag_wt,
n_simulations=10, # Increase number for a better constrained result
run_multicore=FALSE,
verbose=FALSE)


prob_curve <- percentile_from_red_noise(
red_noise = mag_wt_red_noise,
wavelet = mag_wt,
percentile = 0.9)
```

`plot_astro_anchor` *Plot proxy record anchored to an astronomical solution*

Description

Plot the results of the anchoring the extracted signal to an astronomical solution using which was conducted using the `astro_anchor`

Usage

```
plot_astro_anchor(
  astro_solution = NULL,
  proxy_signal = NULL,
  anchor_points = NULL,
  time_dir = TRUE,
  keep_editable = FALSE
)
```

Arguments

`astro_solution` Input is an astronomical solution with with the the proxy record was be anchored to, the input should be a matrix or data frame with the first column being age and the second column should be a insolation/angle/value

proxy_signal
: Input is the proxy data set which will which was anchored to an astronomical solution, the input should be a matrix or data frame with the first column being depth/time and the second column should be a proxy value.

anchor_points
: Anchor points generated using the astro_anchor function

time_dir
: The direction of the proxy record which was assumed during anchoring if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then time_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then time_dir should be set to FALSE `time_dir=TRUE`

keep_editable
: Keep option to add extra features after plotting `Default=FALSE`

Value

The output is a set of 2 plots connected by lines The top plot is the proxy record with anchor points on top of it The bottom plot is the astronomical solution The lines connect the anchor points

Examples

```
# Use the grey_track example tracking points to anchor the grey scale data set
# of Zeeden et al., (2013) to the p-0.5t la2004 solution

grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8
 )

#Use the pretracked grey_track curve which traced the precession cycle
grey_track <- completed_series(
 wavelet = grey_wt,
 tracked_curve  = grey_track,
 period_up  = 1.25,
 period_down  = 0.75,
 extrapolate = TRUE,
genplot = FALSE
)
# Extract precession, obliquity and eccentricity to create a synthetic insolation curve

grey_prec <- extract_signal(
tracked_cycle_curve = grey_track[,c(1,2)],
wavelet = grey_wt,
period_up = 1.2,
period_down = 0.8,
add_mean = FALSE,
tracked_cycle_period = 22,
extract_cycle = 22,
tune = FALSE,
```

```
plot_residual = FALSE
)

grey_obl <- extract_signal(
 tracked_cycle_curve = grey_track[,c(1,2)],
 wavelet = grey_wt,
 period_up = 1.2,
 period_down = 0.8,
 add_mean = FALSE,
 tracked_cycle_period = 22,
 extract_cycle = 110,
 tune = FALSE,
 plot_residual = FALSE
)

grey_ecc <- extract_signal(
 tracked_cycle_curve = grey_track[,c(1,2)],
 wavelet = grey_wt,
 period_up = 1.25,
 period_down = 0.75,
 add_mean = FALSE,
 tracked_cycle_period = 22,
 extract_cycle = 40.8,
 tune = FALSE,
 plot_residual = FALSE
)

insolation_extract <- cbind(grey_ecc[,1],grey_prec[,2]+grey_obl[,2]+grey_ecc[,2]+mean(grey[,2]))
insolation_extract <- as.data.frame(insolation_extract)
insolation_extract_mins <- min_detect(insolation_extract,pts=3)

#use the astrosignal_example to tune to which is an \cr
# ETP solution (p-0.5t la2004 solution).

astrosignal_example <- na.omit(astrosignal_example)
astrosignal_example[,2] <- -1*astrosignal_example[,2]
astrosignal <- as.data.frame(astrosignal_example)

#anchor the synthetic insolation curve extracted from the
# grey scale record to the insolation curve.
#use the anchor_points_grey data set to plot the
#result of using the astro_anchor function

#anchor_points_grey <- astro_anchor(
#astro_solution = astrosignal,
#proxy_signal = insolation_extract,
#proxy_min_or_max = "min",
#clip_astrosolution = FALSE,
#astrosolution_min_or_max = "min",
#clip_high = NULL,
#clip_low = NULL,
#extract_astrosolution  = FALSE,
#astro_period_up  = NULL,
```

```
#astro_period_down  = NULL,
#astro_period_cycle  = NULL,
#extract_proxy_signal  = FALSE,
#proxy_period_up  = NULL,
#proxy_period_down  = NULL,
#proxy_period_cycle  = NULL,
#pts=3,
#verbose=FALSE,
#genplot=FALSE # set verbose to TRUE to allow for anchoring using text feedback commands
#)


plot_astro_anchor(astro_solution = astrosignal,
proxy_signal = insolation_extract,
anchor_points = anchor_points_grey,
time_dir = FALSE,
keep_editable = FALSE)
```

`plot_avg_wavelet` *Plot the average spectral power of a wavelet spectra*

Description

Plot the average spectral power of a wavelet spectra using the results of the `analyze_wavelet` function.

Usage

```
plot_avg_wavelet(
  wavelet = NULL,
  y_lab = "Power",
  x_lab = "period (metres)",
  keep_editable = FALSE
)
```

Arguments

`wavelet`	Wavelet object created using the `analyze_wavelet` function.
`y_lab`	Label for the y-axis `Default="Power"`.
`x_lab`	Label for the x-axis `Default="depth (metres)"`.
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`

Value

The output is a plot of the average spectral power of a wavelet spectra

Examples

```
#Example 1. Plot the average spectral power of the wavelet spectra of
# the Total Solar Irradiance data set of Steinhilber et al., (2012)
TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

plot_avg_wavelet(wavelet=TSI_wt,
                 y_lab= "power",
                 x_lab="period (years)",
                 keep_editable=FALSE)


#Example 2. Plot the average spectral power of the wavelet spectra of \cr
# the magnetic susceptibility data set of Pas et al., (2018)
mag_wt <-
analyze_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10
)
plot_avg_wavelet(wavelet=mag_wt,
                 y_lab= "power",
                 x_lab="period (metres)",
                 keep_editable=FALSE)



#Example 3. Plot the average spectral power of the wavelet spectra of
#the greyscale data set of Zeeden et al., (2013)
grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8
 )

plot_avg_wavelet(wavelet=grey_wt,
                 y_lab= "power",
                 x_lab="period (metres)",
```

```
keep_editable=FALSE)
```

plot_sed_model	*Plot sedimentation modelling results*

Description

The plot_sed_model function is used plot/re-plot the results from the flmw and sum_power_sedrate functions

Usage

```
plot_sed_model(
  model_results = NULL,
  plot_res = 1,
  x_lab = "depth (m)",
  y_lab = "sed rate cm/kyr",
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices"
)
```

Arguments

model_results	Wavelet object created using the analyze_wavelet function
plot_res	Numbers to be used as input form the flmwoutput options 1-8 option 1: slope coefficient, option 2: r squared, option 3: nr of components, option 4: difference to origin, option 5: slope coefficient percentile option 6: r squared percentile, option 7: nr of components percentile, option 8: difference to origin percentile. If the output of the sum_power_sedrate function is used then input should be option 1: sum max power option 2: nr of components
x_lab	Label for x-axis `Default="depth (m)"`
y_lab	Label for y-axis `Default=""sed rate cm/kyr""`
keep_editable	Keep option to add extra features after plotting `Default=FALSE`
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow",

"colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

`color_brewer` Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "`Default=grDevices`

Value

Returns a plot of sedimentation rates vs depth and a value which was generated using the `flmw` or `sum_power_sedrate` functions

Examples

```
#estimate sedimentation rate for the the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018).

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#increase n_simulations to better define the red noise spectral power curve
mag_wt_red_noise <- model_red_noise_wt(wavelet=mag_wt,
n_simulations=10, # increase for a better constrained result
run_multicore=FALSE,
verbose=FALSE)

sedrates <- sum_power_sedrate(red_noise=mag_wt_red_noise,
wavelet=mag_wt,
percentile=0.75,
sedrate_low = 0.5,
sedrate_high = 4,
spacing = 0.05,
cycles = c(2376,1600,1180,696,406,110),
x_lab="depth",
y_lab="sedrate",
run_multicore=FALSE,
genplot = FALSE,
palette_name = "rainbow",
color_brewer= "grDevices",
verbose=FALSE)

plot_sed_model(model_results=sedrates,
plot_res=1,
x_lab = "depth (m)",
y_lab = "sed rate cm/kyr",
```

```
keep_editable=FALSE,
palette_name = "rainbow",
color_brewer= "grDevices")
```

plot_wavelet *Plots a wavelet power spectra*

Description

Plot wavelet spectra using the outcome of the `analyze_wavelet` function.

Usage

```
plot_wavelet(
  wavelet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = TRUE,
  plot_dir = TRUE,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_MTM_peaks = FALSE,
  add_data = TRUE,
  add_avg = FALSE,
  add_pval = FALSE,
  pval_abline = c(0.1, 0.05),
  pval_cutoff = c(0.1),
  add_MTM = FALSE,
  mtm_siglvl = 0.95,
  demean_mtm = TRUE,
  detrend_mtm = TRUE,
  padfac_mtm = 5,
  tbw_mtm = 3,
  plot_horizontal = TRUE
)
```

Arguments

`wavelet`	wavelet object created using the `analyze_wavelet` function.
`lowerPeriod`	Lowest period value which will be plotted
`upperPeriod`	Highest period value which will be plotted
`n.levels`	Number of color levels `Default=100`.
`palette_name`	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
`color_brewer`	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. `"Default=grDevices`
`useRaster`	Plot as a raster or vector image `Default=TRUE`. WARNING plotting as a vector image is computationally intensive.
`periodlab`	Label for the y-axis `Default="Period (metres)"`.
`x_lab`	Label for the x-axis `Default="depth (metres)"`.
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`
`dev_new`	Opens a new plotting window to plot the plot, this guarantees a "nice" looking plot however when plotting in an R markdown document the plot might not plot `Default=TRUE`
`plot_dir`	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE `plot_dir=TRUE`
`add_lines`	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values `Default=NULL`
`add_points`	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values `Default=NULL`
`add_abline_h`	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`

add_abline_v
: Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`

add_MTM_peaks
: Add the MTM peak periods as horizontal lines `Default=FALSE`

add_data
: Plot the data on top of the wavelet `Default=TRUE`

add_avg
: Plot the average wavelet spectral power to the side of the wavelet `Default=FALSE`

add_pval
: add an transparent overlay on the wavelet scalogram based on the p-value and add the p-value curve to the average spectral power curve. The p-value is based on a Monte Carlo simulation of the analyze_wavelet function. The p-value is based on Monte Carlo modelling runs on surrogate data generated based on autocorrelated noise (red noise) the calculated using a windowed (the window is half the size of the data set) temporal autocorrelation and on shuffling the data set resulting in a random data sets which has similar spectral characteristics to the original data set.The shuffling of the data set creates white noise which ensures that high amplitude high frequency/short period cycles do not result in statistical significant peaks. The part of the data generated using the autocorrelated noise (red noise) based on the windowed (the window is half the size of the data set) temporal autocorrelation represent a spectral signature similar to to that of the original data. The original data might include spectral peaks which are the result of astronomical forcing. The result is that the spectral power profile is biased towards rejecting the 0-hypothesis (e.g. no astronomical forcing). By combining the shuffling of the data set with autocorrelated noise a surrogate data set is created which rejects high amplitude high frequency/short period cycles and a reduced biased towards towards rejecting the 0-hypothesis if the data was solely the result of autocorrelated noise. `Default=FALSE`

pval_abline
: add ab-lines to the average spectral power plot which indicate certain p-values `Default=c(0.1,0.5)`

pval_cutoff
: cutoff p-value to be used in the transparent overlay of the wavelet scalogram `Default=c(0.1)`

add_MTM
: Add the MTM plot next to the wavelet plot `Default=FALSE`

mtm_siglvl
: select the significance level (0-1) for the MTM spectrum `Default=0.95`

demean_mtm
: Remove mean from data before conducting the MTM analysis `Default=TRUE`

detrend_mtm
: Remove mean from data before conducting the MTM analysis `Default=TRUE`

padfac_mtm
: Pad factor for the MTM analysis `Default=5`

tbw_mtm
: time bandwidth product of the MTM analysis `Default=3`

plot_horizontal
: plot the wavelet horizontal or vertical eg y axis is depth or y axis power `Default=TRUE`

Value

The output is a plot of a wavelet spectra. if add_MTM_peaks = TRUE then the output of the MTM analysis will given as matrix

Author(s)

Code based on the "analyze.wavelet" and "wt.image" functions of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo (1998). The MTM analysis is from the astrochron R package of Meyers et al., (2012)

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

S.R. Meyers, 2012, Seeing Red in Cyclic Stratigraphy: Spectral Noise Estimation for Astrochronology: Paleoceanography, 27, PA3228, <doi:10.1029/2012PA002307>

Examples

```
#Example 1. A plot of a wavelet spectra using the Total Solar Irradiance
# data set of Steinhilber et al., (2012)

TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

plot_wavelet(
 wavelet = TSI_wt,
 lowerPeriod = NULL,
 upperPeriod = NULL,
 n.levels = 100,
 palette_name = "rainbow",
color_brewer= "grDevices",
 useRaster = TRUE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)",
 keep_editable = FALSE,
```

```
 dev_new=TRUE,
 plot_dir = TRUE,
 add_lines = NULL,
 add_points= NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 add_MTM_peaks = FALSE,
 add_data = TRUE,
 add_avg = TRUE,
 add_pval = FALSE,
 pval_abline = c(0.1,0.05),
 pval_cutoff = c(0.1),
 add_MTM = FALSE,
 mtm_siglvl = 0.95,
 demean_mtm = TRUE,
 detrend_mtm = TRUE,
 padfac_mtm = 5,
 tbw_mtm = 3,
 plot_horizontal=TRUE)

#Example 2. A plot of a wavelet spectra using the magnetic susceptibility
#data set of Pas et al., (2018)
mag_wt <-
analyze_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10
)

plot_wavelet(
wavelet = mag_wt,
lowerPeriod = NULL,
upperPeriod = NULL,
n.levels = 100,
palette_name = "rainbow",
color_brewer= "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
dev_new=TRUE,
plot_dir = TRUE,
add_lines= NULL,
add_points= NULL,
add_abline_h = NULL,
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = TRUE,
add_pval = FALSE,
```

```
pval_abline = c(0.1,0.05),
pval_cutoff = c(0.1),
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal=TRUE)


#Example 3. A plot of a wavelet spectra using the greyscale
# data set of Zeeden et al., (2013)
grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8
 )

plot_wavelet(
wavelet = grey_wt,
lowerPeriod = NULL,
upperPeriod = NULL,
n.levels = 100,
palette_name = "rainbow",
color_brewer= "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
dev_new=TRUE,
plot_dir = TRUE,
add_lines = NULL,
add_points= NULL,
add_abline_h = NULL,
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = TRUE,
add_pval = FALSE,
pval_abline = c(0.1,0.05),
pval_cutoff = c(0.1),
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal=TRUE)
```

plot_win_fft *Plot windowed fft based spectral analysis results*

Description

The plot_win_fft function allows for the (re)plotting of the results of the win_fft

Usage

```
plot_win_fft(
  win_fft = NULL,
  x_lab = c("depth (m)"),
  y_lab = c("frequency cycle/metre"),
  plot_res = 1,
  perc_vis = 0,
  freq_max = NULL,
  freq_min = NULL,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_horizontal = TRUE,
  dev_new = TRUE
)
```

Arguments

win_fft	list which is the results of the win_fft
x_lab	label for the y-axis Default="depth"
y_lab	label for the y-axis Default="sedrate"
plot_res	plot 1 of 8 options option 1: Amplitude matrix, option 2: Power matrix, option 3: Phase matrix, option 4: AR1_CL matrix, option 5: AR1_Fit matrix , option 6: AR1_90_power matrix, option 7: AR1_95_power matrix, option 8: AR1_99_power matrix, Default=1
perc_vis	Cutoff percentile when plotting Default=0
freq_max	Maximum frequency to plot
freq_min	Minimum frequency to plot
keep_editable	Keep option to add extra features after plotting Default=FALSE
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages.

There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

`color_brewer` Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "`Default=grDevices`

`plot_horizontal` plot the wavelet horizontal or vertical eg y axis is depth or y axis power `Default=TRUE`

`dev_new` Opens a new plotting window to plot the plot, this guarantees a "nice" looking plot however when plotting in an R markdown document the plot might not plot `Default=TRUE`

Value

Returns a plot of, which plot 1 of 8 options, option 1: Amplitude matrix option 2: Power matrix option 3: Phase matrix option 4: AR1_CL matrix option 5: AR1_Fit matrix option 6: AR1_90_power matrix option 7: AR1_95_power matrix option 8: AR1_99_power matrix

Examples

```
#Conduct a windowed fft on the magnetic susceptibility record \cr
# of the Sullivan core of Pas et al., (2018).

mag_win_fft <- win_fft(data= mag,
                   padfac = 5,
                   window_size = 12.5,
                   run_multicore = FALSE,
                   genplot = FALSE,
                   palette_name = "rainbow",
                   color_brewer="grDevices",
                   x_lab = c("depth (m)"),
                   y_lab = c("frequency cycle/meter"),
                   plot_res = 1,
                   perc_vis = 0.5,
                   freq_max = 5,
                   freq_min = 0.001,
                   keep_editable=FALSE,
                   verbose=FALSE)

# Plot the amplitude spectra
plot_win_fft(win_fft= mag_win_fft,
```

```
x_lab = c("depth (m)"),
y_lab = c("frequency cycle/meter"),
plot_res = 1,
perc_vis = 0.5,
freq_max = 5,
freq_min = 0.001,
keep_editable=FALSE,
palette_name = "rainbow",
color_brewer="grDevices",
plot_horizontal=TRUE,
dev_new=TRUE)
```

---

plot_win_timeOpt *plot the windowed timeOpt sedimentation rate estimation*

Description

The plot_win_timeOpt function plots a widowed timeOpt sedimentation rate estimation This function is based on the eTimeOptfunction

Usage

```
plot_win_timeOpt(
  win_timeOpt_result = NULL,
  proxy_name = NULL,
  abline_h = NULL,
  abline_v = NULL,
  add_lines = NULL,
  fig_lts = NULL,
  xlab = "depth (m)",
  ylab = "sedrate (cm/kyr)",
  sel_parameter = 3,
  n.levels = 100
)
```

Arguments

win_timeOpt_result
: result of the win_timeOptfunction that needs to be used as input Default=NULL

proxy_name
: the name of the used proxy record Default=NULL

abline_h
: Add horizontal lines to the plot. Specify the lines as a vector e.g. 2,3,5,6 Default=NULL

abline_v
: Add vertical lines to the plot. Specify the lines as a vector e.g. 2,3,5,6 Default=NULL

add_lines
: Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL

fig_lts	Add a text box Default=NULL
xlab	add a label to x-axis Default="depth (m)"
ylab	add a label to y-axis Default="sedrate (cm/kyr)"
sel_parameter	select one of the three returns of the win_timeOptfunction element 1: r_2_envelope matrix element 2: r_2_power matrix element 3: r_2_opt matrix Default=3
n.levels	Number of color levels Default=100.

Value

The output is a plot of the average spectral power of a windowed timeOpt

Author(s)

Based on the eTimeOpt function of the 'astrochron' R package.

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#plot the windowed timeOpt of the magnetic susceptibility record
#of the Sullivan core of Pas et al., (2018).
mag_win_timeOpt <-win_timeOpt(
data = mag,
window_size = 15,
sedmin = 0.1,
sedmax = 1,
numsed = 100,
limit = FALSE,
fit = 2,
fitModPwr = TRUE,
flow = NULL,
fhigh = NULL,
roll = 10 ^ 6,
targetE = c(405.7, 130.7, 123.8, 98.9, 94.9),
targetP = c(20.9, 19.9, 17.1, 17.2),
detrend = TRUE,
normalize =TRUE,
linLog = 1,
run_multicore = FALSE,
verbose=FALSE)

plot_win_timeOpt(win_timeOpt_result = mag_win_timeOpt,
proxy_name= "mag",
abline_h=NULL,
abline_v = NULL,
add_lines=NULL,
fig_lts = NULL,
xlab="depth (m)",
```

```
ylab= "sedrate (cm/kyr)",
sel_parameter=3,
n.levels=100)
```

---

retrack_wt_MC *Re-track cycles using a Monte-Carlo simulation*

Description

When analyzing multi-proxy records an age-model can be created for each proxy. These age-models can be in general agreement but might also indicate conflicting deposition rates. Picking one age-model out of the all multi-proxy age-models and stating that, that age-model is the best overlooks the information contained within the other proxies and hence a degree of error remains the age-model exists.To combine the multiple age-models all the age models can be averaged out and the uncertainty can be calculated by means of the standard deviation. The result is an age-model which takes into account all the age-models from the proxy records. The averaged out age-model does not take into account any small user errors during the creation of the individual age-models nor does the averaging take into account the differences between the age-models and how the initial age-model of a certain proxy might be off in certain intervals. the retrack_wt_MC mitigates these problems by re-tracking periods of astronomical cycles in the wavelet spectra. The re-tracking is based on the information provided by the age-models constructed from the different proxy records. First a synthetic tracked curve is created by adding up fractions (0-1) of the tracked periods of the different proxy records. This synthetic curve is then used to re-track the period/spectral peaks of an astronomical cycle in a randomly select wavelet scalogram. This process is repeated x times. The result x tracked curves which take into account all the original age-models. From the re-tracked curves one can calculate the mean period and the standard deviation. The resulting standard deviation is a good indicator of the quality of the imprint of of astronomical cycles in the proxy records. A small standard deviation indicates that given the input of the different tracked cycles similar periods keep on being tracked indicating the an astronomical is well recorded in the proxy records and as such the age-model is very reliable in set interval. A high standard deviation on the other hand means that the tracking results in vastly different periods of the tracked astronomical cycle, as such the quality of the imprint of the astronomical cycle proxy records is poor and hence the age-model is less-reliable in this interval.

Usage

```
retrack_wt_MC(
  wt_list = NULL,
  data_track = NULL,
  x_axis = NULL,
  smoothing = c("auto"),
  nr_simulations = 50,
  seed_nr = 1337,
  verbose = FALSE,
  genplot = FALSE,
```

```
  keep_editable = FALSE,
  create_GIF = FALSE,
  plot_GIF = FALSE,
  width_plt = 600,
  height_plt = 450,
  period_up = 1.5,
  period_down = 0.5,
  plot.COI = TRUE,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  add_avg = FALSE,
  time_dir = TRUE,
  file_name = NULL,
  run_multicore = FALSE,
  output = 1,
  n_imgs = 50,
  plot_horizontal = TRUE,
  empty_folder = FALSE
)
```

Arguments

`wt_list`	a list containing all the wavelet objects created using the analyze_wavelet wavelet function To create a list use the list function
`data_track`	a matrix containing all the tracked period values. To create the matrix use the cbind function and only add the tracked period values so do not add the depth axis. When combining the tracked period values make sure that all curves have a similar depth spacing.
`x_axis`	The x-axis of the tracked period values
`smoothing`	setting the smoothing parameter and value to either "auto" which uses a automatic loess smoother,"loess" where one can specify Lowess smoothing parameter. or "window" where one can specific the window length of the moving average. one should specify the parameter and its value as vector #' @param wt_list a list containing all the wavelet objects created using the analyze_wavelet wavelet function To create a list use the list function
`nr_simulations`	The number of Monte-Carlo simulations which are to be conducted`Default=50`
`seed_nr`	The seed number of the Monte-Carlo simulations. `Default=1337`
`verbose`	Print text when running the function `Default=FALSE`
`genplot`	Plot a plot with the mean period and + and - standard deviation `Default=FALSE`
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`
`create_GIF`	Create a GIF with the re-tracked lines in the wavelet scalograms `Default=FALSE`
`plot_GIF`	Plot a GIF with the re-tracked lines in the wavelet scalograms`Default=FALSE`
`width_plt`	width of the re-tracked plot `Default=600`

height_plt
: width of the re-tracked plot `Default=450`

period_up
: The period_up parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_up factor is the upper boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks `Default=1.5`,

period_down
: The period_down parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_down factor is the lower boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks `Default=0.5`,

plot.COI
: Option to plot the cone of influence `Default=TRUE`.

n.levels
: Number of color levels `Default=100`.

palette_name
: Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

color_brewer
: Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. `"Default=grDevices`

periodlab
: Label for the y-axis `Default="Period (metres)"`.

x_lab
: Label for the x-axis `Default="depth (metres)"`.

add_avg
: Plot the average wavelet spectral power to the side of the wavelet `Default=FALSE`

time_dir
: The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then time_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then time_dir should be set to FALSE `time_dir=TRUE`

file_name
: Name of the images created using this function. Each file gets a number added to it which corresponds to which number of simulation it was the files are saved in a folder with a similar name created in the current directory

run_multicore Run function using multiple cores `Default="FALSE"`

output #'If output = 1, output is a list which contain 3 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. #'object 2 is a matrix with all the tracked periods. Object 3 is a GIF in which #'all the tracked periods are plotted. If output = 2, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. object 2 is a matrix with all the tracked periods. If output = 3, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4, output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4 output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 5 a matrix with the x-axis and the mean tracked frequency and standard deviation is returned. If output = 6, a matrix with all the tracked periods is returned. If output = 7, a GIF in which all the tracked periods are plotted is returned. `Default=1`

n_imgs Number images used in creating the GIF a high number of images is computationally intensive and will create a large sized GIF `Default=50`

plot_horizontal plot the wavelet horizontal or vertical eg y axis is depth or y axis power `Default=TRUE`

empty_folder Empty the folder in which the images created using this function are saved `Default=FALSE`

Value

The output depends on the output setting If genplot = TRUE a plot will be generated in which the mean period and standard deviation is plotted if plot_GIF = TRUE a GIF with n number of n_imgs will be plotted in which the retraced curve is plotted in a wavelet scalogram If output = 1, output is a list which contain 3 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. object 2 is a matrix with all the tracked periods. Object 3 is a GIF in which all the tracked periods are plotted. If output = 2, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. object 2 is a matrix with all the tracked periods. If output = 3, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4, output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4 output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 5 a matrix with the x-axis and the mean tracked period and standard deviation is returned. If output = 6, a matrix with all the tracked periods is returned. If output = 7, a GIF in which all the tracked periods are plotted is returned

Examples

```
# Re-track the 110kyr eccentricity cycle in the wavelet scalogram
# from the XRF record of the Bisciaro data set of Arts (2014)
```

```
Bisciaro_al <- Bisciaro_XRF[, c(1, 61)]
Bisciaro_al <- astrochron::sortNave(Bisciaro_al,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- astrochron::linterp(Bisciaro_al, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- Bisciaro_al[Bisciaro_al[, 1] > 2, ]

Bisciaro_al_wt <-
 analyze_wavelet(
   data = Bisciaro_al,
   dj = 1 /200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_al_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_al_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_al_wt_track <- completed_series(
#   wavelet = Bisciaro_al_wt,
#   tracked_curve = Bisciaro_al_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_al_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_al_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_ca <- Bisciaro_XRF[, c(1, 55)]
Bisciaro_ca <- astrochron::sortNave(Bisciaro_ca,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- astrochron::linterp(Bisciaro_ca, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- Bisciaro_ca[Bisciaro_ca[, 1] > 2, ]

Bisciaro_ca_wt <-
 analyze_wavelet(
   data = Bisciaro_ca,
   dj = 1 /200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
```

```
    verbose = FALSE,
    omega_nr = 8
  )

# Bisciaro_ca_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_ca_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_ca_wt_track <- completed_series(
#   wavelet = Bisciaro_ca_wt,
#   tracked_curve = Bisciaro_ca_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_ca_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_ca_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE)

Bisciaro_sial <- Bisciaro_XRF[,c(1,64)]
Bisciaro_sial <- astrochron::sortNave(Bisciaro_sial,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- astrochron::linterp(Bisciaro_sial, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- Bisciaro_sial[Bisciaro_sial[, 1] > 2, ]

Bisciaro_sial_wt <-
 analyze_wavelet(
    data = Bisciaro_sial,
    dj = 1 /200 ,
    lowerPeriod = 0.01,
    upperPeriod = 50,
    verbose = FALSE,
    omega_nr = 8
  )

# Bisciaro_sial_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_sial_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
```

```
#
#
# Bisciaro_sial_wt_track <- completed_series(
#   wavelet = Bisciaro_sial_wt,
#   tracked_curve = Bisciaro_sial_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_sial_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_sial_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )


Bisciaro_Mn <- Bisciaro_XRF[,c(1,46)]
Bisciaro_Mn <- astrochron::sortNave(Bisciaro_Mn,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- astrochron::linterp(Bisciaro_Mn, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- Bisciaro_Mn[Bisciaro_Mn[, 1] > 2, ]

Bisciaro_Mn_wt <-
 analyze_wavelet(
   data = Bisciaro_Mn,
   dj = 1 /200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mn_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mn_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mn_wt_track <- completed_series(
#   wavelet = Bisciaro_Mn_wt,
#   tracked_curve = Bisciaro_Mn_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
```

```
#   keep_editable = FALSE
# )
# Bisciaro_Mn_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mn_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_Mg <- Bisciaro_XRF[,c(1,71)]
Bisciaro_Mg <- astrochron::sortNave(Bisciaro_Mg,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- astrochron::linterp(Bisciaro_Mg, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- Bisciaro_Mg[Bisciaro_Mg[, 1] > 2, ]

Bisciaro_Mg_wt <-
 analyze_wavelet(
   data = Bisciaro_Mg,
   dj = 1 /200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mg_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mg_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mg_wt_track <- completed_series(
#   wavelet = Bisciaro_Mg_wt,
#   tracked_curve = Bisciaro_Mg_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_Mg_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mg_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE)
```

```
wt_list_bisc <- list(Bisciaro_al_wt,
               Bisciaro_ca_wt,
               Bisciaro_sial_wt,
               Bisciaro_Mn_wt,
               Bisciaro_Mg_wt)

#Instead of tracking, the tracked solution data sets Bisciaro_al_wt_track,
#Bisciaro_ca_wt_track, Bisciaro_sial_wt_track, Bisciaro_Mn_wt_track,
# Bisciaro_Mn_wt_track and Bisciaro_Mg_wt_track are used

data_track_bisc <- cbind(Bisciaro_al_wt_track[,2],
                     Bisciaro_ca_wt_track[,2],
                     Bisciaro_sial_wt_track[,2],
                     Bisciaro_Mn_wt_track[,2],
                     Bisciaro_Mg_wt_track[,2])

x_axis_bisc <- Bisciaro_al_wt_track[,1]


bisc_retrack <- retrack_wt_MC(wt_list = wt_list_bisc,
             data_track = data_track_bisc,
             x_axis = x_axis_bisc,
             nr_simulations = 20,
             seed_nr = 1337,
             verbose = FALSE,
             genplot = FALSE,
             keep_editable = FALSE,
             create_GIF = FALSE,
             plot_GIF = FALSE,
             width_plt =  600,
             height_plt = 450,
            period_up  =  1.5,
             period_down = 0.5,
             plot.COI = TRUE,
             n.levels = 100,
             palette_name = "rainbow",
             color_brewer = "grDevices",
             periodlab = "Period (metres)",
             x_lab = "depth (metres)",
             add_avg = FALSE,
             time_dir = TRUE,
             file_name = NULL,
             run_multicore = FALSE,
             output = 1,
             n_imgs = 50,
             plot_horizontal = TRUE,
             empty_folder = FALSE)
```

sedrate2tune *Use a sedimentation curve to convert data to the time domain*

Description

Convert a proxy record from the depth to time domain using a sedimentation rate curve

Usage

```
sedrate2tune(
  data = NULL,
  sed_curve = NULL,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

data	Input should be a matrix of 2 columns with first column being depth and the second column is a proxy value
sed_curve	Input should be a matrix of 2 columns with first column being depth and the second column is the sedimentation rate is cm/kyr
genplot	Generates a plot of the proxy record in the time domain `Default=FALSE`.
keep_editable	Keep option to add extra features after plotting `Default=FALSE`

Value

The output is a matrix with 2 columns. The first column is time The second column is the proxy value If `genplot=TRUE` then a time vs proxy value plot will be plotted.

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
# Extract the 405kyr eccentricity cycle from the wavelet scalogram
# from the magnetic susceptibility record of the Sullivan core
# of Pas et al., (2018) and then create a age model using minimal tuning
# (e.g.) set the distance between peaks to 405 kyr. The age model
# (sedimentation rate curve) is then used to convert the data
# from the depth to the time domain
```

```
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)


mag_405 <- extract_signal_stable_V2(
 wavelet = mag_wt,
 period_max = 4,
 period_min = 2,
 add_mean = TRUE,
 plot_residual = FALSE,
 keep_editable = FALSE
)

mag_405_min_tuning <- minimal_tuning(data = mag_405,
pts = 5,
cycle = 405,
tune_opt = "max",
output = 1,
genplot = FALSE,
keep_editable = FALSE)

mag_time <- sedrate2tune(
data=mag,
sed_curve=mag_405_min_tuning,
genplot=FALSE,
keep_editable=FALSE)
```

sum_power_sedrate	*Calculate sum of maximum spectral power for sedimentation rates for a wavelet spectra*

Description

The sum_power_sedrate function is used calculate the sum of maximum spectral power for a list of astronomical cycles from a wavelet spectra. The data is first normalized using the average spectral power curves for a given percentile based on results of the model_red_noise_wt function

Usage

```
sum_power_sedrate(
  red_noise = NULL,
  wavelet = NULL,
  percentile = NULL,
  sedrate_low = NULL,
```

```
  sedrate_high = NULL,
  spacing = NULL,
  cycles = c(NULL),
  x_lab = "depth",
  y_lab = "sedrate",
  run_multicore = FALSE,
  genplot = FALSE,
  plot_res = 1,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  verbose = FALSE
)
```

Arguments

red_noise	Red noise curves generated using the `model_red_noise_wt` function
wavelet	Wavelet object created using the `analyze_wavelet` function
percentile	Percentile value (0-1) of the rednoise runs which is used to normalize the data for. To account for the distribution/distortion of the spectral power distribution based on the analytical technique and random red-noise the data is normalized against a percentile based red-noise curve which is the results of the '`model_red_noise_wt` modelling runs.
sedrate_low	Minimum sedimentation rate (cm/kyr)for which the sum of maximum spectral power is calculated for.
sedrate_high	Maximum sedimentation rate (cm/kyr) for which the sum of maximum spectral power is calculated for.
spacing	Spacing (cm/kyr) between sedimentation rates
cycles	Astronomical cycles (in kyr) for which the combined sum of maximum spectral power is calculated for
x_lab	label for the y-axis `Default="depth"`
y_lab	label for the y-axis `Default="sedrate"`
run_multicore	run simulation using multiple cores `Default=FALSE` the simulation is run at x-2 cores to allow the 2 remaining processes to run background processes
genplot	Generate plot `Default="FALSE"`
plot_res	plot options are 1: sum max power or 2: nr of components `Default=2`
keep_editable	Keep option to add extra features after plotting `Default=FALSE`
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer'

run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

`color_brewer` Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. `"Default=grDevices`

`verbose` Print text `Default=FALSE`.

Value

Returns a list which contains 4 elements element 1: sum of maximum spectral power element 2: number of cycles used in the sum of maximum spectral power element 3: y-axis values of the matrices which is sedimentation rate element 4: x-axis values of the matrices which is depth

If `Default="TRUE"` a plot is created with 3 subplots. Subplot 1 is plot in which the the sum of maximum spectral power for a given sedimentation rate or nr of cycles is plotted for each depth given depth. Subplot 2 is a plot in which the average sum of maximum spectral power is plotted fro each sedimentation Subplot 3 is a color scale for subplot 1.

Author(s)

Based on the asm and eAsm functions of the 'astrochron' R package and the 'eCOCO' and 'COCO' functions of the 'Acycle' software

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Acycle: Time-series analysis software for paleoclimate research and education, Mingsong Li, Linda Hinnov, Lee Kump, Computers & Geosciences,Volume 127,2019,Pages 12-22,ISSN 0098-3004, <doi:10.1016/j.cageo.2019.02.011>

Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing, Mingsong Li, Lee R. Kump, Linda A. Hinnov, Michael E. Mann, Earth and Planetary Science Letters,Volume 501, T2018,Pages 165-179,ISSN 0012-821X,<doi:10.1016/j.epsl.2018.08.041>

Examples

```
#estimate sedimentation rate for the the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018).

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
```

```
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#increase n_simulations to better define the red noise spectral power curve
mag_wt_red_noise <- model_red_noise_wt(wavelet=mag_wt,
n_simulations=10,
run_multicore=FALSE,
verbose=FALSE)

sedrates <- sum_power_sedrate(red_noise=mag_wt_red_noise,
wavelet=mag_wt,
percentile=0.75,
sedrate_low = 0.5,
sedrate_high = 4,
spacing = 0.05,
cycles = c(2376,1600,1180,696,406,110),
x_lab="depth",
y_lab="sedrate",
run_multicore=FALSE,
genplot = FALSE,
plot_res=1,
keep_editable=FALSE,
palette_name = "rainbow",
color_brewer="grDevices",
verbose=FALSE)
```

`track_period_wavelet` *Track the period of a cycle in a wavelet spectra*

Description

Interactively select points in a wavelet spectra to trace a period in a wavelet spectra. The `track_period_wavelet` function plots a wavelet spectra in which spectral peaks can selected allowing one to track a ridge hence one can track the a cycle with a changing period. Tracking points can be selected in the Interactive interface and will be shown as white dots when one wants to deselect a point the white dots can be re-clicked/re-selected and will turn red which indicates that the previously selected point is deselected. Deselecting points can be quite tricky due to the close spacing of points and such the `delpts_tracked_period_wt` can be used to delete points were previously selected using the `track_period_wavelet` function.

Usage

```
track_period_wavelet(
  wavelet = NULL,
  astro_cycle = 405,
```

```
  n.levels = 100,
  track_peaks = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_horizontal = TRUE,
  plot_dir = TRUE,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL
)
```

Arguments

- `wavelet` Wavelet object created using the `analyze_wavelet` function.
- `astro_cycle` Duration (in kyr) of the cycle which traced.
- `n.levels` Number of color levels `Default=100`.
- `track_peaks` Setting which indicates whether tracking is restricted to spectral peaks (track_peaks=TRUE) or whether any point within the wavelet spectra can be selected (track_peaks=FALSE) `Default=TRUE`.
- `periodlab` label for the y-axis `Default="Period (metres)"`.
- `x_lab` label for the x-axis `Default="depth (metres)"`.
- `palette_name` Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: “magma”, “plasma”, “inferno”, “viridis”, “mako”, and “rocket” and “turbo” To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
- `color_brewer` Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "`Default=grDevices`
- `plot_horizontal` plot the wavelet horizontal or vertical eg y axis is depth or y axis power `Default=TRUE`

plot_dir
: The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE `plot_dir=TRUE`

lowerPeriod
: Lowest period value which will be plotted

upperPeriod
: Highest period value which will be plotted

add_lines
: Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values `Default=NULL`

add_points
: Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values `Default=NULL`

add_abline_h
: Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`

add_abline_v
: Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`

Value

Results of the tracking of a cycle in the wavelet spectra is a matrix with 3 columns. The first column is depth/time The second column is the period of the tracked cycle The third column is the sedimentation rate based on the duration (in time) of the tracked cycle

Author(s)

The function is based/inspired on the traceFreq function of the 'astrochron' R package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#Track the 405kyr eccentricity cycle in the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

mag_track <- track_period_wavelet(wavelet = mag_wt,
astro_cycle = 405,
n.levels = 100,
track_peaks = TRUE,
periodlab =  "Period (metres)",
x_lab = "depth (metres)",
```

```
palette_name = "rainbow",
color_brewer = "grDevices",
plot_horizontal = TRUE,
plot_dir = TRUE,
lowerPeriod = NULL,
upperPeriod = NULL,
add_lines = NULL,
add_points = NULL,
add_abline_h = NULL,
add_abline_v = NULL)
```

TSI *Total solar irradiation data (0-9400ka) of steinhilber et al., (2012)*

Description

The Total solar irradiation data set consists of the TSI values of Steinhilber et al., (2012)

Details

Column 1: Age (kyr)
Column 2: Total solar Irradiation (TSI)

References

Steinhilber, Friedhelm & Abreu, Jacksiel & Beer, Juerg & Brunner, Irene & Christl, Marcus & Fischer, Hubertus & Heikkilä, U. & Kubik, Peter & Mann, Mathias & Mccracken, K. & Miller, Heinrich & Miyahara, Hiroko & Oerter, Hans & Wilhelms, Frank. (2012). 9,400 Years of cosmic radiation and solar activity from ice cores and tree rings. Proceedings of the National Academy of Sciences of the United States of America. 109. 5967-71. 10.1073/pnas.1118965109. <doi:10.1073/pnas.1118965109>

---

wavelet_uncertainty *Calculate the uncertainty associated with the wavelet analysis based on the Gabor uncertainty principle*

Description

The wavelet_uncertainty function is used to calculate uncertainties associated with the wavelet analysis based on the Gabor uncertainty principle applied to the continuous wavelet transform using a Morlet wavelet. The calculated uncertainty is the underlying analytical uncertainty which is the result of applying the Gabor uncertainty principle to the continuous wavelet transform using a Morlet wavelet.

Usage

```
wavelet_uncertainty(
  tracked_cycle = NULL,
  period_of_tracked_cycle = NULL,
  wavelet = NULL,
  multi = 1,
  verbose = FALSE,
  genplot_time = FALSE,
  genplot_uncertainty = FALSE,
  genplot_uncertainty_wt = FALSE,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices"
)
```

Arguments

`tracked_cycle` Curve of the cycle tracked using the `track_period_wavelet` function Any input (matrix or data frame) in which the first column is depth or time and the second column is period should work

`period_of_tracked_cycle` period of the tracked curve (in kyr).

`wavelet` wavelet object created using the `analyze_wavelet` function.

`multi` multiple of the standard deviation to be used for defining uncertainty `Default=1`.

`verbose` Print text `Default=FALSE`.

`genplot_time` plot time curves with a upper and lower uncertainty based on Gabor uncertainty principle applied to the continuous wavelet transform using a Morlet wavelet, which uses which uses the omega number (number of cycles in the wavelet) at one standard deviation to define the analytical uncertainty `Default=TRUE`

`genplot_uncertainty` Plot period curves with upper and lower uncertainty based on Gabor uncertainty principle applied to the continuous wavelet transform using a Morlet wavelet, which uses which uses the omega number (number of cycles in the wavelet) to define uncertainty at one standard deviation `Default=TRUE`

`genplot_uncertainty_wt` generate a wavelet plot with the uncertainty based on Gabor uncertainty principle applied to the continuous wavelet transform using a Morlet wavelet superimposed on top of original wavelet plot. The red curve is period of the tracked curve plus the analytical uncertainty. The blue curve is period of the tracked curve min the analytical uncertainty. The black curve is the curve tracked using the '`Default=tracked_cycle_curve` function `Default=TRUE`

`keep_editable` Keep option to add extra features after plotting `Default=FALSE`

`palette_name` Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages.

There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

`color_brewer` Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. `"Default=grDevices`

Value

Results pertaining to the uncertainty calculated based on the Gabor uncertainty principle.
If the genplot_time is TRUE then a depth time plot will be plotted with 3 lines, the mean age,age plus x times the standard deviation and age minus x times the standard deviation .
If the genplot_uncertainty is TRUE then a curve will be plotted with the mean period, the tracked period plus x times the standard deviation and the tracked period minus x times the standard deviation.
If the genplot_uncertainty_wt is TRUE a wavelet spectra will be plotted with the tracked period, the tracked period plus x times the standard deviation,the tracked period minus x times the standard deviation and the area in between will be shaded in grey.

Returns a matrix with 8 columns.
The first column is called "depth" eg. depth
The second column is "period" of the originally tracked period.
The third column is "frequency" of the originally tracked period.
The fourth column "uncertainty in frequency FWHM" is the uncertainty in frequency based on the Gabor uncertainty principle defined as (FWHM) full width at half maximum.
The fifth column "uncertainty in frequency x_times SD" is the uncertainty in frequency based on the Gabor uncertainty principle defined as times x standard deviations.
The sixth column "time mean" is the mean time based on the tracked period.
The seventh column "time plus x_times sd" is the time based on the tracked period plus x times the standard deviation.
The eight column "time min x_times sd" is the time based on the tracked period min x times the standard deviation.

Author(s)

Code based on the "analyze.wavelet" function of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998). The assignment of the standard deviation of the uncertainty of the wavelet is based on the work of Gabor (1946) and Russell et al., (2016)

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Gabor, Dennis. "Theory of communication. Part 1: The analysis of information." Journal of the Institution of Electrical Engineers-part III: radio and communication engineering 93, no. 26 (1946): 429-441.http://genesis.eecg.toronto.edu/gabor1946.pdf

Russell, Brian, and Jiajun Han. "Jean Morlet and the continuous wavelet transform. " CREWES Res. Rep 28 (2016): 115. https://www.crewes.org/Documents/ResearchReports/2016/CRR201668.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#calculate the Gabor uncertainty derived mathematical uncertainty of the
#magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                    wavelet=mag_wt,
#                                    n.levels = 100,
#                                    periodlab = "Period (metres)",
#                                    x_lab = "depth (metres)",
#                                 palette_name="rainbow",
#                                 color_brewer= "grDevices")

#Instead of tracking, the tracked solution data set mag_track_solution is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
```

```
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = FALSE,
  genplot = FALSE,
  keep_editable=FALSE
)

mag_track_complete <- loess_auto(time_series = mag_track_complete,
 genplot = FALSE, print_span = FALSE,keep_editable=FALSE)

uncertainty <- wavelet_uncertainty(
  tracked_cycle = mag_track_complete,
  period_of_tracked_cycle = 405,
  wavelet = mag_wt,
  multi=1,
  verbose = FALSE,
  genplot_time = FALSE,
  genplot_uncertainty = FALSE,
  genplot_uncertainty_wt = FALSE,
  keep_editable=FALSE,
  palette_name="rainbow",
  color_brewer= "grDevices"
)
```

WaverideR *Extracting Signals from Wavelet Spectra*

Description

The continuous wavelet transform enables the observation of transient/non-stationary cyclicity in time-series. The goal of cyclostratigraphic studies is to define frequency/period in the depth/time domain. By conducting the continuous wavelet transform on cyclostratigraphic data series one can observe and extract cyclic signals/signatures from signals. These results can then be visualized and interpreted enabling one to identify/interpret cyclicity in the geological record, which can be used to construct astrochronological age-models and identify and interpret cyclicity in past and present climate systems.

Details

Package: 'WaverideR'

Type: R package

Version: 0.3.2 (begin of 2023)

License: GPL (= 2)

Note

If you want to use this package for publication or research purposes, please cite:

Arts, M.C.M (2023). WaverideR: Extracting Signals from Wavelet Spectra. https://CRAN.R-project.org/package=WaverideR

Author(s)

Michiel Arts

Maintainer: Michiel Arts <michiel.arts@stratigraphy.eu>

References

The 'WaverideR' package builds upon existing literature and existing codebase. The following list of articles is relevant for the 'WaverideR' R package and its functions. Individual articles are also cited in the descriptions of function when relative for set function. The articles in the list below can be grouped in four subjects: (1) Cyclostratigraphic data analysis, (2) example data sets, (3) the (continuous) wavelet transform and (4) astronomical solutions). For each of these categories the relevance of set articles will be explained in the framework of the 'WaverideR' R package.

1. Cyclostratigraphic data analysis

Stephen R. Meyers, Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews,Volume 190,2019,Pages 190-223,ISSN 0012-8252 doi:10.1016/j.earscirev.2018.11.015
The 'astrochron' R package is the most extensive R package with regards to cyclostratigraphic analysis. As such many of the functionalities of the 'WaverideR' R package are #' inspired/based on the 'astrochron' R package. The major difference between #' the 'astrochron' R package and the 'WaverideR' package is that the #' astrochron' R package relies on the Fast Fourier Transform whereas

S.R. Meyers, 2012, Seeing Red in Cyclic Stratigraphy: Spectral Noise Estimation for Astrochronology: Paleoceanography, 27, PA3228, doi:10.1029/2012PA002307
The article of Meyers (2012) explains how the (Multitaper method) MTM technique implemented into The 'astrochron' R package The MTM method can be used to assign confidence levels to spectral peaks and distinguish spectral peaks from harmonic spectral peaks.

Acycle: Time-series analysis software for paleoclimate research and education, Mingsong Li, Linda Hinnov, Lee Kump, Computers & Geosciences,Volume 127,2019, Pages 12-22,ISSN 0098-3004, doi:10.1016/j.cageo.2019.02.011
The 'Acycle' software package is a 'Matlab' based program, which is used for cyclostratigraphic studies. Acycle relies mostly on the Fast Fourier Transform. The 'Coco' and 'eCoco' functions from Acycle formed the inspiration for the flmw sum_power_sedrate functions of the 'Waverider' R package.

Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing, Mingsong Li, Lee R. Kump, Linda A. Hinnov, Michael E. Mann, Earth and Planetary Science Letters,Volume 501, 2018,Pages 165-179, ISSN 0012-821X, doi:10.1016/j.epsl.2018.08.041
Li et al., (2019) introduces the Coco and eCoco functions of the Acycle software package. the 'Coco' and 'eCoco' function of the 'Acycle' software are able to estimate the sedimentation rate based on spectral characteristics of astronomical cycles. The 'Coco' and 'eCoco' function and form the inspiration for the flmw and sum_power_sedrate functions of the 'WaverideR' Package.

Wouters, S., Crucifix, M., Sinnesael, M., Da Silva, A.C., Zeeden, C., Zivanovic, M., Boulvain, F., Devleeschouwer, X., 2022, "A decomposition approach to cyclostratigraphic signal processing". Earth-Science Reviews 225 (103894).doi:10.1016/j.earscirev.2021.103894
Wouters et al., (2022) introduces the Empirical Mode Decomposition (EMD) as part of the 'DecomposeR' R package. EMD is a non-Fast Fourier Transform based spectral analysis technique. The Hilbert transform function inst.pulse of this package is used in WaverideR functions extract_amplitude and Hilbert_transform.

Wouters, S., Da Silva, A.-C., Boulvain, F., and Devleeschouwer, X.. 2021. StratigrapheR: Concepts for Litholog Generation in R. The R Journal. doi:10.32614/RJ2021039
Wouters et al., (2021) introduces the StratigrapheR R package. This package contains functions which format, process, and plot lithologs. The litholog format of Wouters et al., (2021) is used as the standardized input format to convert lithologs to a time series format using the lithlog_disc function. The time series can then be analysed for the imprint of cycles.

#'Huang, Norden E., Zhaohua Wu, Steven R. Long, Kenneth C. Arnold, Xianyao Chen, and Karin Blank. 2009. "On Instantaneous Frequency". Advances in Adaptive Data Analysis 01 (02): 177–229. doi:10.1142/S1793536909000096
The Hilbert transform function inst.pulse of the 'DecomposeR' R package is based on the work of Huang et al., (2009).

Cleveland, W. S. (1979) Robust locally weighted regression and smoothing scatter plots. Journal of the American Statistical Association. 74, 829–836. doi:10.1080/01621459.1979.10481038
Cleveland (1979) explains how the robust locally weighted regression works and how it can be used to smooth data sets. This theory is applied in the loess_auto function of the 'WaverideR' package.

#'Hurvich, C.M., Simonoff, J.S., and Tsai, C.L. (1998), Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion. Journal of the Royal Statistical Society B. 60, 271–293 doi:10.1111/14679868.00125
Hurvich et al., (1998) explains how the Improved Akaike Information Criterion can be used to optimally smooth data sets This theory is applied in the loess_auto function of the 'WaverideR' package.

#'Golub, G., Heath, M. and Wahba, G. (1979). Generalized cross validation as a method for choosing a good ridge parameter. Technometrics. 21, 215–224. doi:10.2307/1268518
Golub et al., (1979) explains how the Generalized cross validation can be used to optimally smooth data sets. This theory is applied in the loess_auto function of the 'WaverideR' package.

2. Example data sets

Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu, Cyclostratigraphic calibration of the Famennian stage (Late Devonian, Illinois Basin, USA), Earth and Planetary Science Letters, Volume 488,2018,Pages 102-114,ISSN 0012-821X, doi:10.1016/j.epsl.2018.02.010
The data set of Pas et al, (2018) is a magnetic susceptibility data measured on the Fammennian aged shales of the from the Illinois basin in the USA. The data set contains the imprint of astronomical cycles in the a Paleozoic succession making it a good example for times (250Ma) when no astronomical solutions are available.

Steinhilber, Friedhelm & Abreu, Jacksiel & Beer, Juerg & Brunner, Irene & Christl, Marcus & Fischer, Hubertus & Heikkilä, U. & Kubik, Peter & Mann, Mathias & Mccracken, K. & Miller, Heinrich & Miyahara, Hiroko & Oerter, Hans & Wilhelms, Frank. (2012). 9,400 Years of cosmic radiation and solar activity from ice cores and tree rings. Proceedings of the National Academy of Sciences of the United States of America. 109. 5967-71. 10.1073/pnas.1118965109. doi:10.1073/

pnas.1118965109
The Total Solar Irradiance record of Steinhilber et al., (2012) is a Holocene record of normalized Total Solar Irradiance in the time domain. The data set is a good example for studying/extracting sub-Milankovitch 5000yr from a relatively (geologically) speaking young record.

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, 10.1016/j.palaeo.2012.11.009
The record of Zeeden et al., (2013) consists of a grey scale record from Miocene sediment cores from offshore Brazil. The record contains a clear imprint of astronomical cycles as such it is a good Neogene example data set to demonstrate the functionalities of the 'WaverideR' R package

3. The (continuous) wavelet transform

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221. Morlet et al., (1982a) together with Morlet et al., (1982b) are the original publications which explain the use of the wavelet to analyse signal.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236. ' Morlet et al., (1982a) together with Morlet et al., (1982b) are the original publications which explain the use of the wavelet to analyse signal.

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf
'Torrence and Compo (1998) shows how the continuous wavelet transform can be used to analyse cyclicity in paleo-climatic data-sets. The equations in this publication forms the basis for many wavelet based packages/software applications.

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet
Gouhier et al., (2021) is the implementation of equations of Torrence and Compo (1998) in the form of the 'biwavelet' R package

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp
Roesch and Schmidbauer et al., (2018) is the article of the 'WaveletComp' R package which is a built upon the functionalities of the 'biwavelet' R package

Russell, Brian, and Jiajun Han. "Jean Morlet and the continuous wavelet transform. " CREWES Res. Rep 28 (2016): 115. https://www.crewes.org/Documents/ResearchReports/2016/CRR201668.pdf
Russell and Han (2016) gives a concise summary of the work of Morlet et al., (1982a) and Morlet et al., (1982b) and the developments since then. The publication also describes how the Gabor uncertainty principle (Gabor 1946) affects the frequency uncertainty of the wavelet which can be used to calculate the analytical uncertainty of a given wavelet spectra.

Gabor, Dennis. "Theory of communication. Part 1: The analysis of information." Journal of the Institution of Electrical Engineers-part III: radio and communication engineering 93, no. 26 (1946): 429-441. http://genesis.eecg.toronto.edu/gabor1946.pdf
Gabor (1946) describes the Gabor uncertainty principle which states how the uncertainty in time and frequency are related in time series analysis.

4. Astronomical solutions

J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B., 2004, A long term numerical solution for the insolation quantities of the Earth: Astron. Astrophys., Volume 428, 261-285. doi:10.1051/00046361:20041335
Laskar et al., (2004) is an astronomical solution which can be used to anchor geological data to absolute ages.

Laskar, J., Fienga, A., Gastineau, M., Manche, H., 2011a, La2010: A new orbital solution for the long-term motion of the Earth: Astron. Astrophys., Volume 532, A89 doi:10.1051/00046361/201116836
Laskar et al., (2011a) is an astronomical solution which can be used to anchor geological data to absolute ages.

Laskar, J., Gastineau, M., Delisle, J.-B., Farres, A., Fienga, A.: 2011b, Strong chaos induced by close encounters with Ceres and Vesta, Astron: Astrophys., Volume 532, L4. doi:10.1051/0004-6361/201117504
Laskar et al., (2011b) is an astronomical solution which can be used to anchor geological data to absolute ages.

J. Laskar,Chapter 4 - Astrochronology,Editor(s): Felix M. Gradstein, James G. Ogg, Mark D. Schmitz, Gabi M. Ogg,Geologic Time Scale 2020,Elsevier,2020,Pages 139-158,ISBN 9780128243602, 'doi:10.1016/B9780128243602.000048
Laskar et al., (2019) explains how astronomical solutions are created and how they should/can be used

Zeebe, Richard E. "Numerical solutions for the orbital motion of the Solar System over the past 100 Myr: limits and new results." The Astronomical Journal 154, no. 5 (2017): 193. doi:10.3847/1538-3881/aa8cce
Zeebe (2017) is an astronomical solution which can be used to anchor geological data to absolute ages.

Richard E. Zeebe Lucas J. Lourens ,Solar System chaos and the Paleocene–Eocene boundary age constrained by geology and astronomy.Science365,926-929(2019) doi:10.1126/science.aax0612
Zeebe and Lourens (2019) is an astronomical solution which can be used to anchor geological data to absolute ages.

Zeebe, R. E. and Lourens, L. J. Geologically constrained astronomical solutions for the Cenozoic era, Earth and Planetary Science Letters, 2022 doi:10.1016/j.epsl.2022.117595
Zeebe and Lourens (2022) is an astronomical solution which can be used to anchor geological data to absolute ages.

`WaverideR_Datasets` *Example data sets for the 'WaverideR' package*

Description

Data sets for testing the 'WaverideR' R package:
The `age_model_zeeden` data set is and age model (anchor points) for the IODP 926 grey scale (154-174m) record of Zeeden et al. (2013)

The astrosignal_example data set consists of pre-generated ETP (eccentricity-tilt-precession) data set based on the p-0.5t la2004 solution and was generated using the etp function of the 'astrochron' R package

The depth_rank_example data set is synthetic succession of sedimentary
The grey data set is the grey scale record of IODP 926 for the interval (154-174m) which originates from Zeeden et al. (2013)

The grey_track data set consists of tracking points of the precession (22 kyr cycle) in the IODP 926 grey scale (154-174m) record of Zeeden et al. (2013)

The mag data set is the magnetic susceptibility record of Pas et al. (2018)

The mag_track_solution is the period of the 405 kyr eccentricity cycle in the magnetic susceptibility record of from Pas et al. (2018)

The TSI data set is the Total Solar Irradiance record of Steinhilber et al. (2012)

The Bisciaro_Mg_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Magnesium (XRF) record of Arts (2014)

The Bisciaro_Mn_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Manganese (XRF)record of Arts (2014)

The Bisciaro_al_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Aluminum (XRF) record of Arts (2014)

The Bisciaro_ca_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Calcium (XRF) record of Arts (2014)

The Bisciaro_sial_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Silicon/Aluminum (XRF) record of Arts (2014)

The Bisciaro_XRF is the XRF data set of Arts (2014)

The anchor_points_Bisciaro_al data set consist of the tie points between the Bisciaro_al record of Arts (2014) and the la2011 solution of laskar et al. (20111)

The GTS_info data set contains the color coding and ages and uncertainties of Geologic Time Scale 2020 of Ogg et al. (2021)

References

Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu, Cyclostratigraphic calibration of the Famennian stage (Late Devonian, Illinois Basin, USA), Earth and Planetary Science Letters, Volume 488,2018,Pages 102-114,ISSN 0012-821X, <doi:10.1016/j.epsl.2018.02.010>

Steinhilber, Friedhelm & Abreu, Jacksiel & Beer, Juerg & Brunner, Irene & Christl, Marcus & Fischer, Hubertus & Heikkilä, U. & Kubik, Peter & Mann, Mathias & Mccracken, K. & Miller, Heinrich & Miyahara, Hiroko & Oerter, Hans & Wilhelms, Frank. (2012). 9,400 Years of cosmic radiation and solar activity from ice cores and tree rings. Proceedings of the National Academy of Sciences of the United States of America. 109. 5967-71. 10.1073/pnas.1118965109. <doi:10.1073/pnas.1118965109>

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

Stephen R. Meyers,Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews,Volume 190,2019,Pages 190-223,ISSN 0012-8252 <doi:10.1016/j.earscirev.2018.11.015>

J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B., 2004, A long term numerical solution for the insolation quantities of the Earth: Astron. Astrophys., Volume 428, 261-285. <doi:10.1051/0004-6361:20041335>

Laskar, J., M. Gastineau, J. B. Delisle, A. Farrés, and A. Fienga (2011b), Strong chaos induced by close encounters with Ceres and Vesta, Astron. Astrophys., 532, L4,<doi:10.1051/0004-6361/201117504>

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

---

win_fft — *Windowed fft based spectral analysis*

Description

The win_fft function for conducts a windowed spectral analysis based on the fft

Usage

```
win_fft(
  data = NULL,
  padfac = 5,
  window_size = NULL,
  run_multicore = FALSE,
  genplot = FALSE,
  x_lab = c("depth (m)"),
```

```
  y_lab = c("frequency cycle/metre"),
  plot_res = 1,
  perc_vis = 0,
  freq_max = NULL,
  freq_min = NULL,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  keep_editable = FALSE,
  verbose = FALSE,
  dev_new = FALSE
)
```

Arguments

data	Input data set should consist of a matrix with 2 columns with first column being depth and the second column being a proxy
padfac	Pad record with zero, zero padding smooths out the spectra
window_size	size of the running window
run_multicore	Run function using multiple cores `Default="FALSE"`
genplot	Generate plot `Default="FALSE"`
x_lab	label for the y-axis `Default="depth"`
y_lab	label for the y-axis `Default="sedrate"`
plot_res	plot 1 of 8 options option 1: Amplitude matrix, option 2: Power matrix, option 3: Phase matrix, option 4: AR1_CL matrix, option 5: AR1_Fit matrix , option 6: AR1_90_power matrix, option 7: AR1_95_power matrix, option 8: AR1_99_power matrix, `Default=1`
perc_vis	Cutoff percentile when plotting `Default=0`
freq_max	Maximum frequency to plot
freq_min	Minimum frequency to plot
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the RColorBrewer::brewer.pal.info() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

color_brewer
: Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "`Default=grDevices`

keep_editable
: Keep option to add extra features after plotting `Default=FALSE`

verbose
: Print text `Default=FALSE`.

dev_new
: Opens a new plotting window to plot the plot, this guarantees a "nice" looking plot however when plotting in an R markdown document the plot might not plot `Default=FALSE`

Value

Returns a list which contains 10 elements element 1: Amplitude matrix element 2: Power matrix element 3: Phase matrix element 4: AR1_CL matrix element 5: AR1_Fit matrix element 6: AR1_90_power matrix element 7: AR1_95_power matrix element 8: AR1_99_power matrix element 9: depth element 10: y_axis If genplot is `Default=TRUE` then a plot of one of the elements 1:8 is plotted

Author(s)

Based on the periodogram function of the 'astrochron' R package.

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#Conduct a windowed ftt on the magnetic susceptibility record
#of the Sullivan core of Pas et al., (2018).

mag_win_fft <- win_fft(data= mag,
                  padfac = 5,
                  window_size = 12.5,
                  run_multicore = FALSE,
                  genplot = FALSE,
                  x_lab = c("depth (m)"),
                  y_lab = c("frequency cycle/metre"),
                  plot_res = 1,
                  perc_vis = 0.5,
                  freq_max = 5,
                  freq_min = 0.001,
                  palette_name ="rainbow",
                  color_brewer= "grDevices",
                  keep_editable=FALSE,
                  verbose=FALSE,
                  dev_new=FALSE)
```

win_timeOpt *Windowed timeOpt sedimentation rate estimation*

Description

The win_timeOpt function for conducts a widowed timeOpt sedimentation rate estimation This function is based on the eTimeOpt but allows for multithreaded analysis speeding up the process of conducting a Windowed timeOpt sedimentation rate estimation

Usage

```
win_timeOpt(
  data = NULL,
  window_size = 10,
  sedmin = 0.5,
  sedmax = 2,
  numsed = 100,
  limit = FALSE,
  fit = 2,
  fitModPwr = TRUE,
  flow = NULL,
  fhigh = NULL,
  roll = 10^6,
  targetE = c(405.7, 130.7, 123.8, 98.9, 94.9),
  targetP = c(20.9, 19.9, 17.1, 17.2),
  detrend = TRUE,
  normalize = TRUE,
  linLog = 1,
  run_multicore = FALSE,
  verbose = FALSE
)
```

Arguments

data	Input data set should consist of a matrix with 2 columns with the first column being depth and the second column being a proxy `Default=NULL`
window_size	size of the moving window in metres `Default=15`
sedmin	Minimum sedimentation rate for investigation (cm/ka). `Default=0.1`
sedmax	Maximum sedimentation rate for investigation (cm/ka). `Default=1`
numsed	Number of sedimentation rates to investigate in optimization grid. `Default=100`
limit	Limit evaluated sedimentation rates to region in which full target signal can be recovered? `.Default=FALSE`
fit	Test for (1) precession amplitude modulation or (2) short eccentricity amplitude modulation? `Default=2`
fitModPwr	Include the modulation periods in the spectral fit? `Default=TRUE`

`flow`	Low frequency cut-off for Taner bandpass (half power point in cycles/ka) `Default=TRUE`
`fhigh`	High frequency cut-off for Taner bandpass (half power point; in cycles/ka) `Default=NULL`
`roll`	Taner filter roll-off rate, in dB/octave. `Default=c(10^6)`
`targetE`	A vector of eccentricity periods to evaluate (in ka). These must be in order of decreasing period, with a first value of 405 ka. `Default= "c(405.7, 130.7, 123.8, 98.9, 94.9)"`
`targetP`	A vector of precession periods to evaluate (in ka). These must be in order of decreasing period. `Default=c(20.9, 19.9, 17.1, 17.2)`
`detrend`	Remove linear trend from data series? `Default=TRUE`
`normalize`	normalize the r2 curves of individual timeOpt runs `Default=TRUE`
`linLog`	Use linear or logarithmic scaling for sedimentation rate grid spacing? (0=linear, 1=log; default value is 1) `Default=1`
`run_multicore`	Run function using multiple cores `Default=FALSE`
`verbose`	print text `Default=FALSE`

Value

Returns a list which contains 10 elements element 1: r_2_envelope matrix element 2: r_2_power matrix element 3: r_2_opt matrix element 4: r_2_envelope_avg element 5: r_2_opt_avg element 6: depth element 7: y_axis element 8: linLog value

Author(s)

Based on the `eTimeOpt` function of the 'astrochron' R package.

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
#Conduct a windowed timeOpt on the magnetic susceptibility record
#of the Sullivan core of Pas et al., (2018).
mag_win_timeOpt <-win_timeOpt(
data = mag,
window_size = 15,
sedmin = 0.1,
sedmax = 1,
numsed = 100,
limit = FALSE,
fit = 2,
fitModPwr = TRUE,
flow = NULL,
fhigh = NULL,
roll = 10 ^ 6,
targetE = c(405.7, 130.7, 123.8, 98.9, 94.9),
targetP = c(20.9, 19.9, 17.1, 17.2),
```

```
detrend = TRUE,
normalize =TRUE,
linLog = 1,
run_multicore =FALSE,
verbose=FALSE)
```

Index